Time-Optimal Control of Closed Quantum Systems

by

Robert Huneault

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Applied Mathematics

Waterloo, Ontario, Canada, 2009

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Recently there has been a lot of interest in the potential applications of performing computations on systems whose governing physical laws are quantum, rather than classical in nature. These quantum computers would have the ability to perform some calculations which would not be feasible for their classical counterparts. To date, however, a quantum computer large enough to perform useful calculations has yet to be built. Before this can be accomplished, a method must be developed to control the underlying quantum systems. This is a problem which can naturally be formulated in the language of control theory. This report outlines the basic control-theoretic approach to time-optimally controlling quantum systems evolving under the dynamics of the Schrödinger operator equation. It is found that under the assumption of non-singularity, the controls which produce time-optimal trajectories are bang-bang. With this in mind, a switching time algorithm is implemented to find optimal bang-bang controls.

Acknowledgements

The creation of this document has been made possible by the constant support and guidance from supervisor Kirsten Morris and former summer research supervisor Joseph Emerson who continued to provide supervision in an unofficial capacity for the entire duration of this project.

I would also like to thank Professor Dong Eui Chang, who provided valuable insight into geometric control and interested me in control theory in the first place. Finally, I would like to acknowledge various contributions from Stephen Vavasis, Brian Ingalls, Ryan Morris, Scott Rostrup, Martin Laforest, Colm Ryan, David Siegel, Scott Bathurst, Alex Shum, Antonio Sanchez, Dhanaraja Khasinathan, Killian Miller, Mark Ilton and Monica Harvey whether or not they realize that they made them.

This research was funded by scholarships from NSERC and OGS.

Contents

List of Tables

List of Figures

List of Notation

Quantum Mechanics

Symbol	Meaning
ψ or ϕ	States of a quantum system – the ket notation is not used
$A^\dagger$	The conjugate transpose of A – the asterisk notation, A^*, is not used
H or $H(t)$	The quantum system Hamiltonian – subscripts denote that the Hamiltonian corresponds to part of the system
$\hbar$	Planck's constant
J	A constant giving the strength of couplings between qubits in a quantum system
q	The number of qubits

Control Theory

Symbol	Meaning
$v(t)$	The control input function
Ω	The set of admissible controls
$X(t)$	An element of the state space of the control system
$\mathcal{X}$	The state space of a control system
$\mathscr{H}$	The Pontryagin Hamiltonian used in the maximum principle
$\lambda(t)$	The costate vector from the maximum principle
$\Lambda(t)$	The costate matrix from the complex matrix maximum principle
m	The number of controls
$\mathcal{J}$	A cost functional from $\mathcal{X} \times \Omega \to \mathbb{R}$

Numerics

Symbol	Meaning
τ	The switching times
ξ	The interval lengths between switching times
α	The matrix of switching times
σ	The stacked columns of α

Lie Groups and Lie Algebras

Symbol	Meaning
$\mathcal{U}(N)$	The set of $N \times N$ unitary matrices
$\mathcal{SU}(N)$	The set of $N \times N$ special unitary matrices
X, Y	An element of $\mathcal{U}(N)$ or $\mathcal{SU}(N)$ (which it is should be clear from the context) – this is not a contradiction with $X(t)$ from control theory since the state space considered here is the set of unitary matrices
$\mathcal{G}$	A general Lie group
g, h	An element of a general Lie group
$\mathfrak{u}(N)$	The set of $N \times N$ skew-Hermitian matrices
$\mathfrak{su}(N)$	The set of $N \times N$ traceless skew-Hermitian matrices
A, B	Elements of $\mathfrak{u}(N)$ or $\mathfrak{su}(N)$
$\{L\}_{LA}$	The Lie algebra generated by the elements of the set L
$\mathcal{L}$	A general Lie algebra
$\mathfrak{g}$, $\mathfrak{h}$, $\mathfrak{f}$	Elements of a general Lie algebra

Miscellaneous

Symbol	Meaning
$\langle \cdot, \cdot \rangle$	An inner product on a Hilbert space
$\mathbb{1}$, $\mathbb{1}_N$	The identity operator and the $N \times N$ identity matrix, respectively
$\cdot \otimes \cdot$	The Kronecker (or tensor) product
$\mathbb{Z}$	A Hilbert space
$\mathbb{C}^n$	An n dimensional complex vector space
$\mathbb{R}^n$	An n dimensional real vector space
$\mathrm{Tr}\{A\}$	The trace of the matrix A
$\mathrm{Re}\{z\}$	The real part of z
i	The imaginary unit
$\mathrm{vec}(\cdot)$	A function which turns a matrix into a vector by stacking the columns
N	The dimension of a square matrix, as in $N \times N$

Subscripts

Symbol	Meaning
$_0$	The value of the variable at the initial time
$_f$	The value of the variable at the final time
$_R$	The real part of a variable
$_I$	The imaginary part of a variable

Chapter 1

Introduction and Motivation

1.1 Scope and Purpose

This report discusses the time-optimal control of systems that arise in the field of quantum computing. While observing the conventions of time-optimal control literature, this report is also intended to be accessible to practitioners of quantum mechanics. Additionally, this report is intended to be accessible to control theorists who have little or no background in quantum mechanics or quantum computing.

For that reason, the most pertinent results and definitions from each field are included for completeness; the more technical details are relegated to the appendix for the interested reader. Standard notation from both fields is used throughout this report except where this is confusing or contradictory. In such cases, a note has been included to explain the choice which has been made. See the list of symbols and conventions for complete details regarding which conventions have been adopted.

1.2 Preliminaries

Quantum Mechanics provides the generally accepted theoretical framework to model systems for which neither classical (Newtonian) nor relativistic classical theories of mechanics provide adequate results. Such systems are typically either on atomic length scales, well-isolated from their environments or both. While this theory has been developed over the last century or so, attention has recently become focused on the possibility of using distinctly quantum systems for computation.

The reason for this interest is that some problems which are prohibitively time consuming on even the fastest classical computers could be solved relatively quickly by computers based on quantum systems. The canonical example is prime factorization. The best known classical algorithms for decomposing integers into their prime factors are variants of the what is called the number field sieve [51]. These

algorithms are known to have a running time which increases faster than any polynomial with the size of the number being factored. This means that the runtime of these algorithms could be on the order of years or decades for large numbers with hundereds of thousands of digits.

In 1994, Peter Shor presented an algorithm [60] which could solve the prime factorization problem in random polynomial time on a computer based on quantum, instead of classical, systems. This means that, in theory, numbers which are not possible to factor in a reasonable amount of time on classical computers would be relatively easy to factor on a quantum computer. The principle that allows this speedup is called superposition. Loosely speaking, this is the ability of a quantum system to be in more than one state at the same time.

Since the publication of Shor's paper, a variety of other potential applications of quantum computing have been discovered and investigated; Grover's algorithm, for example, provides a quadratic speedup over classical algorithms for searching an unsorted database [25]. At the present time, however, a quantum computer with enough power to perform meaningful computations has yet to be built. Before this can happen, there are a number of technical and theoretical hurdles which must be overcome.

With regards to building a useful quantum computer, Raymond Laflamme, head of the Institute for Quantum Computing at the University of Waterloo, has said,

> "The first step is to understand quantum properties; the second, to learn to control them; the third is to use them for something interesting." [1]

The second step mentioned leads to questions which can be naturally posed in the language of control theory; that is the focus of this report.

With a few low-dimensional exceptions, solutions to the types of control problems which arise from quantum computing systems tend to be complicated and very difficult to represent analytically. As a result, quantum control problems often lead to numerical algorithms for determining solutions.

The remainder of this report is organized as follows: Chapter 2 introduces quantum mechanics and quantum computing, explains the physics behind the system we are considering, and discusses some of the existing literature; Chapter 3 is devoted to finding conditions which guarantee that the system is controllable; Chapter 4 introduces Pontryagin's Maximum Principle, a major tool in optimal control, then generalizes it for complex matrix systems and, finally, applies the Principle to a general quantum system as well as the more specific one qubit system; numerical methods for computing time-optimal quantum controls and the results from runing this algorithm are presented in Chapters 5 and 6; and Chapter 7 summarizes the entire thesis and provides directions for future work.

Chapter 2

System and Problem Statement

2.1 Quantum Mechanics and Computing

We now introduce the basic theory from quantum mechanics (QM) and quantum computing (QC) required to understand the remainder of this report. The knowledgeable practitioner of QM and QC may skip this section unless they wish to familiarize themselves with the notation.

This overview of the basic results is not intended to be complete but to give enough background to understand the system with which we will be working. As a result, the logical flow may differ somewhat from standard texts in the field. For a more general and complete introduction to QM and QC than the scope of this report allows see [48].

The Postulates of Quantum Mechanics

The mathematics of quantum mechanics are based heavily on linear algebra. Quantum mechanics textbooks typically use the Dirac notation, where vectors are represented by "ket" brackets (such as $|\psi\rangle$.) In this thesis, we stick to the standard linear algebra notation and omit the brackets (so the previous vector is just written as ψ.)

The postulates of quantum mechanics give the theoretical framework from which we can connect the physical quantum systems to mathematical representations. The postulates do not tell us anything about specific quantum systems, but rather give insight into general traits possessed by all quantum systems. The first postulate describes the space on which a quantum system evolves.

Postulate 1. (Quantum State Space) *Associated to any isolated physical system is a complex Hilbert space which is known as the* state space *of the quantum system. The system is completely describable by a unit vector in the system's state space.*

Remark 2.1. Postulate 1 does not specify which state space should be used or of what dimension this space might be. These are properties which are determined empirically on a case by case basis.

One important example for quantum computing is the state space of a 2-level (or 2-dimensional) quantum system. The quantum "spin" of an electron (or of an atom) is such a system. Any quantum state of the system can be described by the vector

$$\psi = \alpha\psi_{up} + \beta\psi_{down}$$

where α and β are complex scalars satisfying $|\alpha|^2 + |\beta|^2 = 1$ (because the quantum state must be a unit vector.) The vectors ψ_{up} and ψ_{down} are basis vectors of the quantum state space respresenting the spin up and spin down states respectively.

In an analagous classical system, a state would have to be one of the basis states, either ψ_{up} or ψ_{down}. In quantum mechanics, linear combinations of the basis states are possible. This ability to be in more than one state at the same time is called *superposition*. As mentioned in the introduction, this is the property which allows for computational speedups in quantum computers. The next postulate shows how to represent quantum systems which are composed of more than one smaller system.

Postulate 2. (Composite Systems) *The state space of a composite physical system is the kronecker (or tensor) product of the state spaces of the component physical systems.*

For example, if we have n systems with states ψ_i, then the joint state of the total system is

$$\Psi = \psi_1 \otimes \cdots \otimes \psi_n$$

Postulate 2 covers, for example, the case of a system with two or more interacting spin systems. Before stating the third postulate, we introduce some important definitions.

Definition 2.1. (Adjoint Operator [18, A.3.57]) *Let $\mathbb{Z}$ be a Hilbert space with inner product $\langle , \rangle$ and let Q be a linear operator on that Hilbert space. There exists a unique linear operator $Q^\dagger$ such that*

$$\langle Qx, y\rangle = \left\langle x, Q^\dagger y\right\rangle,$$

for all x and y in Z. The operator $Q^\dagger$ is called the adjoint operator of Q on the Hilbert space $\mathbb{Z}$.

Theorem 2.1. *Consider a finite Hilbert space $\mathbb{Z}$ where the elements are operators represented by $N \times N$ matrices. The standard inner product on this space is*

$$\langle X, Y\rangle = \mathrm{Tr}(\bar{X}^T Y).$$

In this case, the adjoint of an operator is equivalent to the conjugate transpose of that operator.

Proof. Let X and Y be elements of $\mathbb{Z}$ and Q be on operator acting on $\mathbb{Z}$.

$$\begin{aligned}\langle QX, Y\rangle &= \mathrm{Tr}\left(\overline{(QX)}^T Y\right)\\ &= \mathrm{Tr}(\bar{X}^T\bar{Q}^T Y)\\ &= \left\langle X, \bar{Q}^T Y\right\rangle\end{aligned}$$

By Definition 2.1 it follows that

$$Q^\dagger = \bar{Q}^T.$$

□

We henceforth use the dagger notation (†) to represent conjugate transposition in finite-dimensional matrix spaces. Note also that we are using the quantum mechanics convention of applying the conjugate transpose to the first argument of the inner product and not the second as is done in mathematics.

Definition 2.2. (Unitary Operator [18, A.3.62]) *A unitary operator, X, is a bounded linear operator acting on a Hilbert space, $\mathbb{Z}$. Unitary operators satisfy*

$$X^\dagger X = XX^\dagger = \mathbb{1}.$$

where $\mathbb{1}$ is the identity operator.

Remark 2.2. In finite-dimensional space, unitary operators are represented by $N \times N$ matrices and the identity operator, $\mathbb{1}$, by the $N \times N$ identity matrix. The set of $N \times N$ unitary matrices is denoted by $\mathcal{U}(N)$.

In most quantum references, the standard notation for a unitary matrix or operator is U. In pure mathematics, however, the notation for the set of all unitary matrices of dimension $N \times N$ is $\mathrm{U}(N)$. To make matters even more confusing, the letter u is typically reserved for control input functions in control theory.

To avoid confusion, this report will use the symbol X for a unitary operator or matrix and $\mathcal{U}(N)$ for the set of all unitary matrices of dimension $N \times N$. The symbol v will be used for controls.

Definition 2.3. (Hermitian Operator [18, A.3.68]) *A Hermitian operator, H, is a bounded operator acting on a Hilbert space, $\mathbb{Z}$, satisfying*

$$\langle Hx, y\rangle = \langle x, Hy\rangle, \quad \forall\, x, y \in \mathbb{Z},$$

where $\langle , \rangle$ is the inner product defined on the Hilbert Space. In mathematics this is called a self-adjoint *operator.*

Remark 2.3. In finite-dimensional space, Hermitian operators are represented as $N \times N$ matrices and satisfy

$$H^\dagger = H.$$

Similarly, a matrix S with the property that

$$S^\dagger = -S,$$

is called *skew-Hermitian*. The set of $N \times N$ skew-Hermitian matrices is denoted by $\mathfrak{u}(N)$, a notation which should become clear during the discussion of Lie algebras and Lie groups.

Remark 2.4. The problems considered in this report are exclusively finite-dimensional. As a result, all Hermitian and unitary operators can be expressed as matrices.

The final postulate we state in this thesis is the most directly useful for control theory and tells us how quantum systems evolve.

Postulate 3. (Evolution) *The evolution of a closed quantum system is described by a unitary transformation. That is, the state ψ at time t_1 is related to the state ψ' at time t_2 by a unitary operator X which depends only on the times t_1 and t_2,*

$$\psi' = X\psi$$

The dynamics of the quantum state of a closed quantum system are described by the Schrödinger equation*,*

$$i\hbar\frac{d\psi}{dt} = H\psi, \tag{2.1}$$

where H is a Hermitian operator and $\hbar$ is Planck's constant*.*

Remark 2.5. There is a one-to-one correspondence between the unitary transformation picture of quantum evolution and the Schrödinger equation view [48]. Solutions of (2.1) have the form $\psi(t) = \exp(-\frac{i}{\hbar}Ht)\psi_0$, where it is easy to show that $\exp(-\frac{i}{\hbar}Ht)$ is a unitary operator. Furthermore, any unitary operator can be expressed as $\exp(iH)$ for some Hermitian operator H.

We can also express the dynamics of the system in terms of the dynamics of the unitary operator by

$$i\hbar\frac{dX(t)}{dt} = HX(t). \tag{2.2}$$

This equation is known as the *Schrödinger operator equation* and is very useful for quantum computing as it closely relates to the idea of quantum logic gates, which we discuss in Section 2.1. The following definition and theorems pertain to solutions of (2.2).

Definition 2.4. (Right Invariant [4]) *Let $\mathcal{S}$ be a dynamical operator system. Let the time-varying operator $s(t)$ be a solution of $\mathcal{S}$ and let M be an arbitrary, constant operator. The system $\mathcal{S}$ is called right invariant if $s(t)M$ is also a solution of $\mathcal{S}$ for any such M.*

Theorem 2.2. *The Schrödinger operator equation, (2.2), is right invariant.*

Proof. Take the derivitive of $X(t)M$ to get

$$\begin{aligned}\frac{dX(t)M}{dt} &= \frac{dX(t)}{dt}M \\ &= -iHX(t)M \\ &= -iH\left(X(t)M\right),\end{aligned}$$

which shows that $X(t)M$ is also a solution of (2.2). □

The following theorem is the specialization of a more general result from [4, §1]. The statement in [4] is presented without proof and is written in terms of Lie groups (see Chapter 3).

Theorem 2.3. *If the initial state $X(0)$ of system (2.2) is a unitary operator, then the solution, $X(t)$, remains unitary for all time.*

Proof. At time $t = 0$, $X^\dagger X = \mathbb{1}$, since $X(0)$ is unitary. We now show that $X^\dagger X$ remains constant for all $t \geq 0$.

$$\begin{aligned}\frac{dX^\dagger X}{dt} &= \frac{dX^\dagger}{dt}X + X^\dagger\frac{dX}{dt} \\ &= \left(\frac{dX}{dt}\right)^\dagger X + X^\dagger\frac{dX}{dt} \\ &= \left(\frac{-i}{\hbar}HX\right)^\dagger X + X^\dagger\frac{-i}{\hbar}HX \\ &= \frac{i}{\hbar}X^\dagger H^\dagger X - \frac{i}{\hbar}X^\dagger HX \\ &= \frac{i}{\hbar}X^\dagger HX - \frac{i}{\hbar}X^\dagger HX \\ &= 0,\end{aligned}$$

where we have used the properties of the conjugate transpose operation (see Appendix A) as well as equation (2.2). We therefore conclude that $X^\dagger X = \mathbb{1}$ for all time.

An identical calculation shows that $XX^\dagger = \mathbb{1}$ for all time and so X remains unitary under the dynamics of the Schrödinger operator equation (2.2). □

Computing Basics

We now move on to give a brief introduction to some very basic principles of computing theory. Please refer to [40] and [48] for a more complete account. Most of this description can be found in those references. Since quantum computing is

analagous in many ways to classical computing, we begin the discussion with an outline of some definitions from the field of classical computing.

Classical computers consist essentially of wires to transport information and logic gates to manipulate that information. The wires and logic gates behave according to the laws of classical mechanics, specifically, the laws of electricity and magnetism.

The basic unit of information in the classical setting is a bit. A bit may only take the values 0 or 1 and may be combined with other bits into sequences. Computations are performed by operating on the bits with logic gates (such as AND, NOT, OR, etc...). In physical computers, the value of a bit is stored as voltage differences in an electronic circuit.

To be able to perform arbitrary computations on data, we need to be able to perform arbitrary manipulations of the corresponding sequence of bits. As a result, it is very useful, not to mention practical, to be able to construct arbitrary bit manipulations with a finite (and hopefully reasonably small) set of logic gates. Such a set of gates is called *universal*.

In quantum computing, information is stored and manipulated in physical systems which are modelled by the laws of quantum mechanics. In this context we still have the analogues of wires and logic gates, though their implementation is slightly different.

The basic unit of quantum information is the qubit (short for *qu*antum *bit*). Physically, a qubit is a 2-level quantum system. "Sequences" of qubits are interacting two-level systems and, by Postulate 2, mathematically represented by a tensor product of those component systems. The dynamics of the total system are described by the Schrödinger equation.

There are several ways to physically implement a qubit, all of which involve interacting 2-level systems. Often the systems are spin systems. Such spins could come from trapped ions or even the atoms in a molecule. The latter method is the basis for the nuclear magnetic resonance (NMR) implementations of quantum computing.

Unlike the classical case, we see from the evolution postulate that manipulating a quantum state is the same as acting on it with a unitary operator. These unitary operators are therefore the analogues of the classical logic gates and are indeed called *quantum logic gates* when used for this purpose. Much in the same way as in classical computing, we want a finite set of quantum logic gates to perform arbitrary manipulations of information or, a universal quantum gate set. The focus of this report, however, is on producing (or synthesizing) given quantum logic gates and not on generating a universal gate set.

2.2 Unitary Synthesis as a Control Problem

We now examine the question of how we can produce a given quantum logic gate. This is equivalent to producing the unitary X from the quantum evolution equation

$$\psi_1 = X\psi_2.$$

The dynamics of X are given by the Schrödinger operator equation (2.2), so the problem is therefore one of driving the system from the identity at time zero, to the desired unitary, denoted X_d, at some final time. This is the type of problem dealt with by the field of control theory.

Before we can really talk about control however, we need to know a little bit more about the system itself. Specifically, we have not yet introduced any means by which we may influence the system to exert our desired control actions. We have also not seen how the system would evolve on its own in the absence of control input. All of this information is encoded in the Hermitian operator H. H is called the *system Hamiltonian*.

Most quantum systems that could be useful for computation possess several common properties. The systems of interest have a Hamiltonian of the form

$$H = H_{int} + H_{ext}.$$

H_{int} is called the *drift Hamiltonian* or the *internal Hamiltonian*. Physically, it represents the interaction between the qubits in the system. As such, it describes the internal dynamics of the quantum system and gives its behaviour in the absence of control. H_{ext} is called the *control Hamiltonian* and represents the part of the system Hamiltonian which can be affected in some way by a control input. This is responsible for the system's behaviour in response to some external stimulus, or control. Some examples of physical stimuli which affect quantum systems are radio-fields and lasers. The actual physical control used depends on the implementation of the quantum computer.

Both H_{int} and H_{ext} must themselves be Hermitian operators. We will only consider the case where H_{int} is time independent. The control Hamiltonian on the other hand will not be constant and the time-dependence is introduced by a control input function, $v(t)$. Our overall system takes the form

$$i\hbar\frac{dX}{dt} = \left[H_{int} + H_{ext}(v)\right] X.$$

The exact operators H_{int} and H_{ext} must be determined empirically. We assume that these are known for the systems we are working with.

A common structure for the system Hamiltonian is

$$H(t) = H_0 + \sum_{k=1}^{m} H_k v_k(t). \tag{2.3}$$

This has the same form as previously mentioned with $H_{int} = H_0$ and $H_{ext} = \sum H_k v_k(t)$. We will consider only Hamiltonians of this form in this report.

The problem we address for the remainder of this report is that of driving the system

$$-i\hbar\dot{X}(t) = \left[H_0 + \sum_{j=1}^{m} H_j v_j(t)\right] X(t)$$

from the initial state $X(0) = \mathbb{1}$ to a desired final state X_d.

A more detailed look at the physics behind a quantum computing system can be found in section II of [63], which refers specifically to nuclear magnetic resonance implementations of quantum computers.

2.3 Decoherence and Other Limitations

Recall that the postulates of quantum mechanics given previously are for closed (or isolated) systems. This means that, to a good approximation, the quantum system does not interact with its environment.

When a quantum system interacts with its environment it ceases to be specifically quantum in nature since it is no longer isolated. This process is known as *decoherence*. A quantum system which has experienced significant decoherence loses the computing advantage it once had over classical systems since, for all intents and purposes, it has turned into a system which can be descibed by classical mechanics. From a computing point of view, this is highly undesirable. Every quantum system will eventually interact with its environment and experience decoherence. The more time over which the system evolves, the more the effects of decoherence will become evident.

A possible way to minimize this effect is to look for time-optimal solutions to the control problem that is, solutions to the problem which can be achieved in the minumum possible time. This is the approach used in this report.

A practical consideration which is not addressed in this report is the problem of *crosstalk*. It is assumed that every control may selectively address a particular qubit, or set of qubits. When this assumption is not completely valid, we say that crosstalk is occuring between the controls. Many systems will exhibit this effect to some degree, particularly when multiple control signals are being applied simultaneously [63].

Another problem that arises with quantum systems is that of measurement. This is related to the concept of observability from control theory. Control systems are often viewed in terms of input-output relations. For quantum systems, outputs are tricky to deal with because they require measurement which could potentially disturb the state of the system. Some quantum systems show promise for continuous feedback controls via weak measurement. The system would be stochastic in nature

and it would need to take into account the back action resulting from performing measurements. See references [10] and [41]. To avoid this issue, we consider only open-loop control. This means that no measurements are taken while the system is being controlled.

The lack of feedback makes the control system more susceptible to instrument error. Since any solutions found to this control problem are only useful in as much as that may be physically implemented on a real quantum experiment, understanding the quirks of the particular setup would be very important. Physical systems will never be as neat as the model equation we are using. We ignore this problem but note that it is something that would need to be considered in the future.

2.4 Existing Literature

The field of optimal control theory has existed in earnest since the 1950's. A major result, the Maximum Principle of Pontryagin et al. [52], was translated to English from Russian in 1962 and gives necessary conditions for a control to be optimal (see Chapter 4). Other results such as the higher order maximum principle of Krener [42] in 1977 also have applications to the quantum control problem presented in the previous section.

Related geometric control theory results have existed since the 1970's. Results originally designed to deal with control of rigid bodies under rotation such as [12] were extended to more general systems which could even encompass the types of models used to describe quantum systems since, as it turns out, there are many structural similarities between the two. This structure was used in [29] to examine the question of controllability (discussed in Chapter 3) and in [61] to examine what could be said about the possible forms of optimal controls on these systems. A standard reference in the field of geometric control is the text by Jurdjevic [28].

While these theoretical results have been around for awhile, their actual application to quantum systems is relatively new. For example, it was only in a survey paper from 1993 by Warren et al. [65] where it was finally declared that the dream of coherent quantum control was alive. This paper was written before the widespread interest in quantum computing began in 1994 (as a result of Shor's prime factorization algorithm) and, therefore, does not mention potential applications to this field.

A reason for the delay between the existence of the theory and its application as discussed in [65], is that it was only around this time that the physical control systems became reliable enough to be used for the purpose. The experimental progress in the field of active laser control demonstrated the possibily of being able selectively break particular bonds in polyatomic molecules. The authors of [65] briefly discusses the application of optimal control theory to designing control strategies to achieve this objective.

Related to the breaking of chemical bonds is the more general idea of quantum state control. Given a particular quantum state ϕ_0, we may wish to drive this to another quantum state ϕ_f. Such a problem is considered in [67] and also references therein. In this case, the author looks for time-optimal solutions. Using what is known as the Born approximation, the author finds that for bounded controls, the solutions are bang-bang (Bang-bang controls are discussed in Chapter 4.4.)

Another application of quantum control deals with what is called the "population transfer" between energy levels in a n-level quantum system. This typically involves the excitation of ions using active laser control. Boscain *et al* examines this problem in a series of papers [5, 6, 8].

The approach used by Boscains involves both exploiting the structure of the space on which the problem evolves and the use of the Pontryagin's Maximum Principle to find energy optimal controls (these are control laws which minimize the total energy used to synthesize the desired unitary.) In low dimensional cases ($n = 2$ and $n = 3$) some controls are found analytically. It is also shown that abnormal extremals cannot be optimal in these cases (see Chapter 4.4 for a discussion of normal and abnormal controls).

The same problem is considered by Chang and Sepulchre in [16], but with a time-optimal cost instead of an energy optimal one. Additionally, this paper uses the symmetry inherent in the problem to reduce the complexity and the resulting control laws are found to be bang-bang. The authors suggest that generalizing the result to n-level systems would be possible.

Yet another approach to quantum control is discussed by Carlini *et al* in [14, 15]. These papers discuss the time-optimal evolution and time-optimal synthesis of a unitary operator from the calculus of variations perspective. The papers find Hamiltonians which can produce a desired quantum state or unitary operator in the shortest time. No attempt is made to match this approach to a specific type of physical control experiment although, in practice, the form of the Hamiltonian would be restricted by the physical setup.

The most closely related literature to the problem we are considering is that which deals with the optimal control of a unitary operator. Much of the work which has been done in this area considers Nuclear Magnetic Resonance (NMR) implementations of quantum computing since, to date, these have been the easiest to work with physically. NMR quantum computers use ensembles of polyatomic molecules which contain atoms with odd numbers of nucleons (protons and neutrons and both nucleons), such as carbon-13. Such atoms are spin 1/2 particles. The spins of the various atoms are coupled, which is to say that they interact as described by the internal Hamiltonian. The strength of each coupling is given by a constant. Electromagnetic fields of specific frequencies are able to, to a good approximation, selectively induce particular rotations in the spin of a particular qubit. For a more thorough review of NMR quantum computing, refer to [49], [54] and [63].

Since 2000, D'Alessandro and collaborators have produced several papers dealing with optimal control of quantum systems, two of which are [20] and [22]. Both of

these papers are motivated by the application of synthesizing unitary operators as logic gates for NMR quantum computing applications. They approach the problem by finding energy-optimal solutions while allowing for the possibility of unbounded controls.

The second of the two papers, [20] ignores the drift term for a two qubit system, but computes a bound on the error associated with doing so. It is found that minimizing this error is equivalent to solving a time-optimal control problem. The time-optimal problem is solved on the space $\mathcal{SO}(4)$ (a component of the decomposition of $\mathcal{SU}(4)$) and, as already mentioned, the drift term is ignored. This leads to a characterization of normal and abnormal extremals for the problem.

Published around the same time, [53], finds decompositions of $\mathcal{SU}(2)$, which it is suggested would be useful for computing time-optimal controls. Also around the same time, [68] looks at the one qubit version of the time-optimal control problem. This is essentially the same problem examined in Chapter 4.6 of this report, though the paper does not provide the proofs of the theorems stated within.

Boscain et al., previously mentioned for their work in population transfer between energy levels, turn their attention to synthesizing unitary operators in [7] and [9]. The former paper looks at a space which is isomorphic to a one qubit system and investigates the number of switches required to produce a desired unitary operator. The latter paper shows that time-optimal solutions for the particular system are bang-bang and compare the results with those of Khaneja from [33].

A series of papers by Khaneja *et al*, including [37], [33], [34], [35], [36], and [69], treat a variety of problems associated with synthesizing a unitary operator. The first of these papers [33] deals specifically with finding time-optimal control of quantum spin systems to produce unitary operators. The results rely on the fact that the control signals which may be generated in NMR implementation of quantum computing are orders of magnitude larger than the magnitudes of the couplings in the internal Hamiltonian. The authors assume the controls can be made arbitrarily large; in effect, the controls are considered to be unbounded.

Unlike many control problems, unbounded controls do not trivialize the problem because not every state can be reached in arbitrarily short time. A demonstration of this is given in [29]. The results of this paper include that the fastest way to get from one unitary operator to another reduces to a geometric problem of finding the fastest way to get between points in the quotient space of the system. The optimal controls are found to be of the form pulse-drift-pulse, where the pulses are control inputs of large amplitude and the drift component corresponds to natural evolution of the system.

Also from Khaneja and collaborators, [38] and [58] present the mathematical framework for an algorithm as well as the algorithm itself (GRAPE) which can be used to numerically compute controls for quantum spin systems. This algorithm will be discussed in further detail in Chapter 5.

Many of the results in the field of quantum control and optimization have been summarized in the recent textbooks [21] and [50]. The former reference also provides

an introduction to quantum mechanics and controllability of quantum systems. However, the time-optimal control of a unitary operator with bounded controls, which is the focus of this report, is not given much attention therein.

2.5 Time-Optimal Control

As in many of the references listed in the previous section, we approach the problem by trying to find time-optimal ways to generate unitary operators. The reason for this is that we want to minimize the effects of noise and decoherence. Recall that decoherence is an unmodelled distortion of the quantum system whose effects tend to build over time. Therefore, the desired unitary operators should be synthesized as efficiently as possible to minimize these effects [6, 33]. Generating the desired unitary operators in the shortest time should, in principle, minimize the effects of decoherence and other noise which are always present.

In some cases, such as the NMR system considered in [33], the relative magnitudes of the control fields which can be applied to the system allow us to neglect the drift effect from the internal part of the Hamiltonian. In reality, every control system will have some physical limitations when implemented. In this report, we consider systems with limitations on the maximum control signal that can be applied to the system. In other words, we consider systems which have bounded control inputs. The control inputs $v_k(t)$ for our system will have the form

$$v_k(t) \in [v_{min}, v_{max}],$$

where v_{min} is the smallest possible control field we can produce and v_{max} is the largest. Since the control function typically represents the amplitude of some form of electromagnetic radiation, we choose v_{min} to be zero. We note that despite not allowing negative controls, the controllability properties of the system remain unchanged, as is demonstrated in Chapter 3.

The time-optimality problem can be stated as follows. Of all the available controls, $v(t)$, which drive the system (2.2) from the initial condition $X(0) = \mathbb{1}$ to the desired final condition $X(t_f) = X_d$, which is the one which does so in the minimum time, t_f. This is called a *time-optimal* control problem. The standard framework used for dealing with this problem is Pontryagin's Maximum Principle from [52], as we shall see more formally in Chapter 4.

2.6 Non-dimensionalizing the Problem

Before getting into the more technical details, we rewrite the problem rescaling the time variable t. The dynamics are given by

$$-i\hbar \frac{dX(t)}{dt} = H(v)X(t).$$

where $v_j(t) \in [0, u_{max}]$.

We can rescale the time variable by an arbitrary positive constant γ by introducing $\hat{t}$ defined as

$$\hat{t} = \gamma \frac{t}{\hbar}.$$

This gives

$$\frac{dt}{d\hat{t}} = \frac{\hbar}{\gamma}. \tag{2.4}$$

By the chain rule, we see that

$$\frac{dX(t)}{d\hat{t}} = \frac{dX(t)}{dt}\frac{dt}{d\hat{t}}.$$

The two previous equations give

$$\frac{dX(t)}{dt} = \frac{\gamma}{\hbar}\frac{dX(t)}{d\hat{t}}.$$

Equating this to the original dynamics gives that

$$i\hbar\frac{\gamma}{\hbar}\frac{dX(t)}{d\hat{t}} = H(v)X(t),$$

which simplifies to

$$i\gamma\frac{dX(t)}{d\hat{t}} = H(v)X(t).$$

Multiplying both sides by $-\frac{i}{\gamma}$, we get

$$\frac{dX(t)}{d\hat{t}} = -\frac{i}{\gamma}H(v)X(t).$$

We finally define $X(t) = \hat{X}(\hat{t})$ and set γ to one, which gives us the Schrödinger operator equation as

$$\frac{dX}{dt} = -iH(v)X(t),$$

where the hats have been dropped for simplicity. This is the form of the equation we consider for the remainder of the report unless otherwise stated. We note that γ is not useful in analysis but that it sometimes becomes important to rescale the equation when solving it numerically (see Chapter 5). Additionally, we may wish to scale out constants such as v_{max}. When $H(v)$ is a linear function of v (as it will be for all examples in this thesis), both of these can be done by choosing an appropriate value of γ. For example, the latter can be done by setting $\gamma = v_{max}$ and defining a new control function by

$$\hat{v}_j(t) = \frac{v_j(t)}{\gamma}.$$

This allows us to consider bounded controls of the form

$$\hat{v}_j(t) \in [0, 1].$$

Chapter 3

Controllability

In this chapter we discuss the issue of controllability. Informally, a system is controllable if it is possible to drive that system to any final state from any initial state with a valid control input. In other words, controllability means that it is actually possible to control the system in the desired way. Controllability results for linear systems are well known and can be stated in terms of the rank of what is called the *controllability matrix*. This matrix may easily be constructed for a given linear system (see [47, §2.2]). For nonlinear systems there is no general approach for establishing controllability.

Despite this theoretical shortcoming, however, the structure of the quantum systems with which we are working allows us to determine their controllability properties. In particular, we show that these quantum systems evolve on Lie groups. Consequently, results from the fields of Lie groups and Lie algebras are used to find the required controllability properties. The much cited 1972 paper by Sussman and Jurdjevic [29] gives criteria for controllability of systems evolving on Lie groups. The results from that paper are extended in [2], Altafini's related 2002 paper.

3.1 Basic definitions

Quantum States and Unitary Operators

Recall from Chapter 2 that there are two forms of the Schrödinger equation - the quantum state formulation given by

$$|\dot{\psi}(t)\rangle = -iH\,|\psi(t)\rangle\,, \tag{3.1}$$

and the operator forumlation given by

$$\dot{X}(t) = -iHX(t). \tag{3.2}$$

While we may formulate a control problem for either version of the equation, the physical interpretations of the two systems are different. Recall that a transformation from one quantum state to another is given by Postulate 3 and is a unitary transformation of the form

$$\psi_1 = X\psi_2$$

for some unitary operator, X.

In the quantum state version (3.1) the time evolution of the system is carried by the quantum state, $\psi(t)$. By contrast, the time evolution in the operator version, (3.2), is carried by the unitary operator $X(t)$. In (3.1), we are interested in transforming a particular initial quantum state into a particular final quantum state. In (3.2), we are not interested in the specific quantum state but only in the *transformation* between initial and final quantum state.

The term *quantum state control* is used to refer to the control of system (3.1), while *quantum operator control* or just *operator control* is used to refer to the control of system (3.2). The application determines which version should be used. In quantum computation, it is almost always important to know about the transformation between arbitrary quantum states, and not about any specific quantum state. As a result, we consider operator control exclusively for the remainder of this thesis.

Admissible Controls and State Space

For a general control system, we define the control input $v(t)$. For any real t, $v(t)$ is an element of $\mathbb{R}^m$. We also define the set of admissible controls Ω. The set Ω contains all the allowable control functions. For example, we are considering systems with bounded controls. This admissible set is defined by

$$\Omega_{\text{bounded}} = \left\{ \begin{array}{c} \text{Piecewise continous functions } v(t) \text{ in } \mathbb{R}^m \text{ such that} \\ v_{min} \leq v_l(t) \leq v_{max} \text{ where } t \geq 0 \text{ and } l = 1, \ldots, m \end{array} \right\}.$$

In the general case, a function $v(t)$ that is an element of Ω is called an *admissible control*.

We also define the state space $\mathcal{X}$. This is the space over which the control system evolves. Elements of $\mathcal{X}$ will be denoted by X. Note the distinction between a quantum state (say ψ) and the state of a control system which are not the same thing in general. For example, when we are considering operator control, the state space is the set of unitary matrices or, in other words, the state space is the set of transformations from one quantum state to another. In the case of quantum state control, however, the quantum state and the state of the control system happen to coincide. Henceforth, the term *state* refers to the state of the control system, while the term *quantum state* refers to the state of the quantum system.

Controllability and Reachable Sets

The definition of controllability can be formalized in several ways. We use the following definition for this report which is stated for a nonlinear affine system in [57]. The quantum systems we are considering are a special case of this type of system.

Definition 3.1. (Controllability [57, §11]) *A system is controllable if for any two points X_0 and X_f in $\mathcal{X}$, there exists a control v in Ω and a finite, non-negative time T such that for $X(0) = X_0$ we have that $X(T) = X_f$.*

Controllability is a property that we wish our control system to possess. The remainder of this chapter is devoted to finding conditions which guarantee controllability and tests for determining the same. A related notion is that of reachability which is defined in terms of the *reachable set*.

Definition 3.2. (Reachable Set [57, §11]) *A* reachable set *from a given point X_0 is the set*

$$R^U(X_0, T) \in \mathcal{X}$$

of points which are attainable from $X(0) = X_0$ in time T by applying an admissible control and such that the trajectories $X(t)$ remain inside of U, which is some neighbourhood of X_0. We use the notation

$$R(X_0, T)$$

to denote the case where $U = \mathcal{X}$ and the notation

$$R(X_0)$$

to denote the case where we consider all points that are attainable for some finite time T.

Reachability from X_0 means that $\mathcal{R}(X_0) = \mathcal{X}$ or in words, that the reachable set from X_0 is the entire state space. If the existence of an admissible control to drive X_0 to X_f only happens in the limit as T tends to infinity for some X_0 and X_f then we use the terms *approximately controllable*, *approximately reachable set* and *approximate reachability from* X_0.

Remark 3.1. While reachability from a point and controllability are equivalent for linear systems, the same is not true for nonlinear systems. For example, even if it is possible to drive the system from some initial point to every point in the state space it may still not be possible to do so from any given initial point in the state space, which means the system would not be controllable. On the other hand, a system which is reachable from every point in the state space is controllable by definition.

Universal Quantum Gate Sets

Recall that the unitary operators we are working with are to be used as logic gates for a quantum computer. We notice that the idea of a universal gate set which was introduced in Section 2.1 bears some similarity to the idea of controllability.

Definition 3.3. (Universal Gate Set [48]) *A finite set of logic gates (or unitary operators) is called* universal *for quantum computation if any desired unitary operator may be approximated to arbitrarily high accuracy using a finite sequence of logic gates from that set. A finite set of unitary operators has the* universality *property if it is universal.*

Intuitively, one might suspect that having a universal gate set would imply that arbitrary unitary operators could be achieved and therefore that the system must be controllable. This is not necessarily the case, however. Controllability and universality are fundamentally different concepts in that controllability is a property of the control system, while universality is a property of a set of operators which happen to be elements of the state space.

Specifically, consider a quantum system evolving on the set unitary matrices such as

$$\dot{X} = -iH(v)X.$$

Controllability implies that starting from any given unitary matrix, we may produce any other unitary matrix by applying some control input, v. Universality implies that we may find a finite set of unitary matrices, M_U, from which it is possible to generate to arbitrary accuracy any other unitary matrix with a product of the form

$$M_1 M_2 \dots M_n$$

where each M_j is an element of M_U.

One might still wonder if universality implies controllability or vice-versa. If a system is controllable, then it is certainly possible to produce any finite set of unitary matrices. The question of whether any such set can be universal cannot be answered by controllability. However, it is stated in [43] that almost any individual quantum logic gate forms a universal set. (The proof is in [66].) As such, controllability does happen to imply the ability to generate a universal gate set, though not without some help.

We also note that the existence of a universal gate set does not imply controllability. It is certainly possible to have a universal gate set which contains an element which cannot be reached using an admissible control input $v(t)$. Even if every element of some universal gate set may be achieved by applying an admissible control, this is not sufficient to conclude that the system is controllable because a universal gate set need only be able to produce arbitrarily good approximations of every unitary matrix while controllability requires that every matrix be exactly reached in finite time. As such, the existence of a universal gate set is only enough to imply approximate controllability. To demonstrate controllability, more theory is required.

3.2 Lie Groups and Lie Algebras

Differential geometry and, in particular, Lie groups and Lie algebras are very important for studying controllability in many applications. For that reason, a brief summary of the key definitions and results in this field is presented below with examples. For a more complete introduction to Lie theory, refer to [17], [19] or [55].

Lie Algebras

Lie algebras are vector spaces on which has been defined an additional "bracket" operation as defined below.

Definition 3.4. (Lie Algebra [17, §II]) *A* Lie algebra $\mathcal{L}$ *over a field* $\mathcal{F}$ *is a vector space over* $\mathcal{F}$ *with an additional binary operation* $\mathcal{L} \times \mathcal{L} \to \mathcal{L}$. *This operation associates with an ordered pair of elements* $\{\mathfrak{g}, \mathfrak{h}\}$ *in* $\mathcal{L}$ *an element* $[\mathfrak{g}, \mathfrak{h}]$ *and is required to satisfy the following properties:*

1. *Bilinearity:*

$$[\mathfrak{g} + \mathfrak{h}, \mathfrak{f}] = [\mathfrak{g}, \mathfrak{f}] + [\mathfrak{h}, \mathfrak{f}], \quad [\mathfrak{g}, \mathfrak{h} + \mathfrak{f}] = [\mathfrak{g}, \mathfrak{h}] + [\mathfrak{g}, \mathfrak{f}],$$

$$[\alpha\mathfrak{g}, \mathfrak{h}] = [\mathfrak{g}, \alpha\mathfrak{h}] = \alpha[\mathfrak{g}, \mathfrak{h}], \quad \forall \alpha \in \mathcal{F}.$$

2. *An element bracketed with itself is zero:*

$$[\mathfrak{g}, \mathfrak{g}] = 0.$$

3. *Jacobi identity:*

$$[\mathfrak{g}, [\mathfrak{h}, \mathfrak{f}]] + [\mathfrak{h}, [\mathfrak{f}, \mathfrak{g}]] + [\mathfrak{f}, [\mathfrak{g}, \mathfrak{h}]] = 0.$$

Definition 3.5. (Lie Bracket [21, §3.1], [55, §1.6]) *A binary operation which satisfies properties (1-3) in Definition 3.4 is called a* Lie bracket.

Note that in some references the Lie bracket is written as $\{,\}$. A trivial example of a Lie algebra is any vector space on which we define the Lie bracket by $[\mathfrak{g}, \mathfrak{h}] = 0$ for all $\mathfrak{g}$ and $\mathfrak{h}$ in the vector space. A less trivial example is the vector space $\mathbb{R}^3$ with the standard vector cross-product. The cross-product can easily be verified to satisfy properties of a Lie bracket. For quantum mechanics, the most useful Lie bracket is the *commutator*.

Theorem 3.1. *For operators* $\mathfrak{g}$ *and* $\mathfrak{h}$*, the commutator, defined by*

$$[\mathfrak{g}, \mathfrak{h}] = \mathfrak{g}\mathfrak{h} - \mathfrak{h}\mathfrak{g}$$

satisfies properties (1-3) and is therefore a Lie bracket.

Proof. We first demonstrate bilinearity of the commutator.

$$\begin{aligned}[\mathfrak{g}+\mathfrak{h},\mathfrak{f}] &= (\mathfrak{g}+\mathfrak{h})\mathfrak{f}-\mathfrak{f}(\mathfrak{g}+\mathfrak{h}) \\ &= \mathfrak{g}\mathfrak{f}+\mathfrak{h}\mathfrak{f}-\mathfrak{f}\mathfrak{g}-\mathfrak{f}\mathfrak{h} \\ &= (\mathfrak{g}\mathfrak{f}-\mathfrak{f}\mathfrak{g})+(\mathfrak{h}\mathfrak{g}-\mathfrak{f}\mathfrak{h}) \\ &= [\mathfrak{g},\mathfrak{f}]+[\mathfrak{h},\mathfrak{f}].\end{aligned}$$

The proof for linearity of the second argument is identical. We also have that

$$\begin{aligned}[\alpha\mathfrak{g},\beta\mathfrak{h}] &= \alpha\mathfrak{g}\beta\mathfrak{h}-\beta\mathfrak{h}\alpha\mathfrak{g} \\ &= \alpha\beta(\mathfrak{g}\mathfrak{h}-\mathfrak{h}\mathfrak{g}) \\ &= \alpha\beta[\mathfrak{g},\mathfrak{h}],\end{aligned}$$

which proves that the commutator is bilinear. The condition that $[\mathfrak{g},\mathfrak{g}]=0$ is trivial to show. Finally, we see that

$$\begin{aligned}[\mathfrak{g},[\mathfrak{h},\mathfrak{f}]]+[\mathfrak{h},[\mathfrak{f},\mathfrak{g}]]+[\mathfrak{f},[\mathfrak{g},\mathfrak{h}]] &= \mathfrak{g}(\mathfrak{h}\mathfrak{f}-\mathfrak{f}\mathfrak{h})-(\mathfrak{h}\mathfrak{f}-\mathfrak{f}\mathfrak{h})\mathfrak{g} \\ &+ \mathfrak{h}(\mathfrak{f}\mathfrak{g}-\mathfrak{g}\mathfrak{f})-(\mathfrak{f}\mathfrak{g}-\mathfrak{g}\mathfrak{f})\mathfrak{h} \\ &+ \mathfrak{f}(\mathfrak{g}\mathfrak{h}-\mathfrak{h}\mathfrak{g})-(\mathfrak{g}\mathfrak{h}-\mathfrak{h}\mathfrak{g})\mathfrak{f} \\ &= 0\end{aligned}$$

which proves the Jacobi identity for the commutator. Therefore the commutator is a Lie Bracket by definition. □

Theorem 3.2. *The vector space of $N\times N$ skew-Hermitian matrices $\mathfrak{u}(N)$ (introduced in Remark 2.3) forms a Lie algebra with the commutator as the Lie bracket.*

Proof. It can be easily verified that the set of skew-Hermitian matrices are a vector space. Since the commutator is a Lie bracket, it is sufficient to show that the vector space of skew-Hermitian matrices is closed under this operation. Indeed we have that

$$\begin{aligned}-[A,B]^\dagger &= -(AB-BA)^\dagger \\ &= -(B^\dagger A^\dagger - A^\dagger B^\dagger) \\ &= -(BA-AB) \\ &= AB-BA \\ &= [A,B],\end{aligned}$$

where we have used properties of the conjugate transpose operation which can be found in Appendix A. This shows that the bracket operation is closed for any skew-Hermitian matrices A and B and therefore the skew-Hermitian matrices are a Lie algebra under this commutator. □

As with many algebraic objects, Lie algebras may have Lie subalgebras.

Definition 3.6. ([56, §1.4]) *A Lie subalgebra is a subspace of a Lie algebra $\mathcal{L}$ which is closed under the Lie bracket.*

Theorem 3.3. *The subspace $\mathfrak{su}(N)$ of $\mathfrak{u}(N)$ defined by*

$$\mathfrak{su}(N) = \{A \in \mathfrak{u}(N), \text{ where } \mathrm{Tr}(A) = 0\}$$

is a Lie subalgebra of $\mathfrak{u}(N)$.

Proof. We need to show that this subspace is closed under the Lie bracket. This means we must show that for any A and B in $\mathfrak{su}(N)$, the element $[A, B]$ has zero trace. The trace of the element $[A, B]$ is

$$\mathrm{Tr}([A, B]) = \mathrm{Tr}(AB - BA) = \mathrm{Tr}(AB) - \mathrm{Tr}(BA) = 0,$$

by the cyclic property of trace. Therefore $\mathfrak{su}(N)$ is closed under the Lie bracket and forms a Lie subalgebra of $\mathfrak{u}(N)$ □

When determining whether or not a system is controllable, it is important to know if a given subset of elements from the Lie algebra "generates" the remaining elements.

Definition 3.7. *We say that a (possibly infinite) set of elements $L = (L_1, L_2, \dots)$ from a Lie algebra $\mathcal{L}$* generates *the Lie algebra if*

$$\{L_1, L_2, \dots\}_{LA} = \mathcal{L}.$$

The notation $\{\cdot\}_{LA}$ denotes the set of elements which may be produced by repeatedly taking Lie brackets of the elements inside the curly braces. This includes any element of the form

$$[A_1, [A_2, [A_3, \dots]]],$$

where $A_1, A_2, \dots$ are all elements of the set L (the A_is need not be unique.)

Lie Groups and Lie Subgroups

The complete definition of a Lie group is technical and beyond the scope of this thesis. A simplified version of the definition is included below for completeness, but is not necessary for understanding the rest of the chapter. The interested reader may consult Appendix B for further details.

Definition 3.8. (Lie Group [21, §3.1]) *A* Lie group *is a group which is also an analytic differentiable manifold and whose group operations $g * h$ and g^{-1} are analytic.*

The proof of the following theorem is in Appendix B.

Theorem 3.4. *([17]) The unitary matrices (introduced in Section 2.1) form a Lie group.*

Just as with Lie algebras, Lie groups may have Lie subgroups. Informally, a subset $\mathcal{H}$ of a Lie group $\mathcal{G}$ is a subgroup if it is both a subgroup and a submanifold of $\mathcal{G}$. It is known that a closed subgroup of a Lie group is a Lie subgroup [17, §V.XIV]. In quantum computing, the special unitary group, $\mathcal{SU}(N)$, is a very important Lie subgroup of the unitary group.

Definition 3.9. (Special Unitary Group) *The special unitary group is defined as*

$$\mathcal{SU}(N) = \{X \in \mathcal{U}(N) \text{ where } \det(X) = 1\}$$

that is, the special unitary group contains the elements of the unitary group with unit determinant.

Relation Between Lie Algebras and Lie Groups

Lie groups and Lie algebras are very closely related. In fact, there is a finite-dimensional Lie algebra associated to each Lie group, although the converse is not true. Simply connected and connected Lie groups are completely determined by their associated Lie algebras, which reduces the study of these groups to the study of their Lie algebras [64, §3].

The question then becomes how to determine which Lie algebras are associated to which Lie groups. Answering this question involves technical details beyond the scope of this report. For the purposes of finite quantum systems (for which we will only be dealing with matrix groups) we state that the Lie algebra $\mathcal{L}$ associated to a Lie group $\mathcal{G}$ is the set of all the matrices L for which $\exp(tL)$ is an element of $\mathcal{G}$. In other words, the matrix exponential maps elements of the Lie algebra onto elements of the Lie group.

It is known that the matrix exponential of a skew-Hermitian matrix is a unitary matrix and also that any unitary matrix can be written as the exponential of a skew-Hermitian matrix. Thus we may accept that the Lie algebra of skew-Hermitian matrices have as an associated Lie group, the unitary matrices.

From [29] we have a useful way of expressing any element of a Lie group in terms of the generators of the underlying Lie algebra.

Theorem 3.5. *Let $\mathcal{G}$ be any connected group which has $L = (L_1, \ldots, L_n)$ as the generators of its associated Lie algebra. Every element of $\mathcal{G}$, denoted by g, can be expressed in the form*

$$g = \exp(t_1 A_1) \exp(t_2 A_2) \ldots \exp(t_m A_m), \tag{3.3}$$

where each A_i is an element of L.

Proof. Consider a connected Lie group $\mathcal{G}$. Let $\mathcal{L}$ be the Lie algebra of $\mathcal{G}$ and let $\mathcal{L}$ be generated by

$$L = (L_1, \ldots, L_n)\,.$$

We will consider the set $\mathcal{G}'$ that is composed of all finite products of elements of the form $\exp(tL_i)$ where L_i is in L and t is a real number. We intend to show that $\mathcal{G} = \mathcal{G}'$.

A space, S, is called *path-connected* if there exists a continuous function, $f : \mathbb{R} \to S$ defined on $[0, 1]$ for which $f(0) = s_1$ and $f(1) = s_2$ for any elements s_1 and s_2 in S.

The set $\mathcal{G}'$ is clearly path-connected since we can define the continuous function f in terms of the exponentials $\exp(tL_i)$ (it may be necessary to rescale t.) It is also easy to show that $\mathcal{G}'$ satisfies the properties of a group. We now use [29, Lemma 2.2] which states that a path-connected subgroup of a Lie group must be a Lie subgroup.

The Lie group $\mathcal{G}'$ contains the subgroups generated by the elements of L by its definition. It must therefore hold that the elements of L are in the Lie algebra associated to $\mathcal{G}'$. It now follows that $\mathcal{G}' = \mathcal{G}$ since they have the same Lie algebra.

We have shown that any element of the Lie group $\mathcal{G}$ may be expressed as a finite product of exponentials of the generators of the associated Lie algebra which completes the proof. □

The previous result tells us that generators of a Lie algebra also generate the associated Lie group via products of their exponentials.

We use the notation

$$\exp(\mathcal{L})$$

to denote the set of all possible elements generated by the product (3.3). This allows us to write the relation between a Lie group $\mathcal{G}$ and a Lie algebra $\mathcal{L}$ as

$$\exp(\mathcal{L}) = \mathcal{G}. \tag{3.4}$$

Finally, it is known that sub-algebras of Lie algebras have a one-to-one correspondance to subgroups of their associated Lie group. We have already seen that skew-Hermitian matrices form the Lie algebra associated with the Lie group of unitary matrices. In the notation of (3.4), this can be written as

$$\exp(\mathfrak{u}(N)) = \mathcal{U}(N).$$

It can also be shown that the Lie algebra of skew-Hermitian matrices with zero trace have the special unitary group as their associated Lie group, or that

$$\exp(\mathfrak{su}(N)) = \mathcal{SU}(N).$$

3.3 Structure of the Problem

The reason that Lie algebras and Lie groups are important in quantum control is related to the way quantum systems evolve. Closed quantum systems evolve under

unitary transformation. This means, as we have already seen, that any two states in $\mathcal{X}$ can be related by the equation

$$\psi_2 = X\psi_1,$$

where X is a unitary operator depending only on the initial and final times. We also recall that by Theorem 2.3 the dynamics given by

$$\dot{X} = -iH(v)X \tag{3.5}$$

preserve the unitary evolution of the system or, in other words, that X remains a unitary operator under these dynamics for all time.

It is therefore clear that the evolution of the quantum system occurs on a state space which is also $\mathcal{U}(N)$, the Lie group of $N \times N$ unitary matrices, where the parameter N depends on the size of the system.

Furthermore, $-iH(v)$ is always an element of the $N \times N$ skew-Hermitian matrices, $\mathfrak{u}(N)$, which is the Lie algebra associated to the unitary group. We will see that the Lie algebra generated by the span of $-iH(v)$ over the admissible controls Ω plays a major role in determining the reachable set of a point.

We note that many implementations of quantum computing consider systems which evolve on the Lie subgroup of special unitary matrices, introduced in Definition 3.9, as opposed to the group of unitary matrices. The reason for this is that any two quantum states ψ and ϕ are considered equivalent if they are related by

$$\psi = e^{i\theta}\phi$$

for some real θ. In this case we say that ψ and ϕ are equivalent up to the *global phase factor* $e^{i\theta}$. This equivalence makes sense because the global phase factor cannot be measured and so, from the observational viewpoint, does not affect the physical properties of the system. See [48, §2] for more information.

The group of special unitary matrices can be thought of as the unitaries from which the ambiguity of the global phase factor has been removed [13, §18]. As a result it is not necessary for us to consider these phase factors if we only look at systems which evolve on $\mathcal{SU}(N)$. It was proved in Theorem 2.3 that the Schrödinger operator equation (3.5) preserves unitarity. The more general statement of this theorem in [4] allows us to say that if the initial state is an element of $\mathcal{SU}(N)$ and the set of Hamiltonians $\{H_0, \dots H_m\}$ are all in $\mathfrak{su}(N)$, then a solution $X(t)$ of (3.5) remains in $\mathcal{SU}(N)$ for all time. That is, if the Hamiltonians $\{H_0, \dots H_m\}$ are all in the Lie algebra associated to $\mathcal{SU}(N)$ then the evolution of (3.5) will be on the subgroup $\mathcal{SU}(N)$ instead of on $\mathcal{U}(N)$.

3.4 Admissible Controls

We use many results from [29] in this report. All of these results are stated in terms of three types of admissible control sets; the unrestricted controls, restricted controls

($|v_j(t)| \leq 1$) and bang-bang controls ($v_j(t) = \pm 1$). We wish to consider a slightly different set of admissible controls. The purpose of this section is to demonstrate that our quantum control system can be written in a form which makes the result from [29] applicable.

Recall that the control inputs often correspond to the amplitude of some electromagnetic radiation. For that reason we choose an admissible control set of the form

$$\Omega = \left\{ \begin{array}{c} \text{Piecewise continous functions } v(t) \text{ in } \mathbb{R}^m \text{ such that} \\ 0 \leq v_j(t) \leq v_{max} \text{ where } t \geq 0 \text{ and } j = 1, \ldots, m \end{array} \right\}. \tag{3.6}$$

where v_{max} is a positive real number. This is the same as the set of bounded controls mentioned previously except with $v_{min} = 0$.

This choice of admissible control set is not equivalent to any of those used in [29]. We give a simple example where the restricted controls from [29] give different controllability results than the admissible set we wish to use.

Consider a one-dimensional system evolving on the real line with dynamics

$$\dot{x} = v.$$

With controls restriced by $|v| \leq 1$, the system is clearly controllable since any element of the state space can eventually be reached by applying the control $v(t) = -1$ or $v(t) = 1$, regardless of what the initial state of the system may be. On the other hand, with controls restricted by $0 \leq v \leq 1$ the system is no longer controllable because it is not possible to reach any state for which x is less than the initial state x_0.

For a quantum system of the form

$$\dot{X} = -i \left(H_0 + \sum_{j=1}^{m} H_j v_j \right) X,$$

with admissible controls given by (3.6), we can rewrite the system in terms of a restricted control as follows. Let $v_k = \frac{v_{max}}{2}(\tilde{v}_k + 1)$, where $|\tilde{v}_k| \leq 1$. We have that

$$\dot{X} = -i \left(H_0 + \sum_{j=1}^{m} H_j \left(\frac{v_{max}}{2} (\tilde{v}_{j+1}) \right) \right) X.$$

A few lines of algebra gives that

$$\dot{X} = -i \left(H_0^* + \sum_{j=1}^{m} H_j^* \tilde{v}_j \right) X,$$

where $H_0^* = H_0 + \sum_{j=1}^{m} H_j^*$ and $H_k^* = \frac{v_{max}}{2} H_j$.

We note that $-iH_0^*$ is skew-Hermitian since it is the sum of skew-Hermitian matrices. The operator $-iH_0^*$ therefore remains in $\mathfrak{su}(N)$ after rewriting the problem

in this form. This shows that we may write the quantum system as an equivalent system, preserving the Lie group and Lie algebra structure while allowing us to use results proven for controls of the form $|v_j| \leq 1$. Furthermore, it is easy to show that the Lie algebra generated by the sets $\{H_j\}$ and $\{H_j^*\}$ (for $j = 0 \ldots m$) are the same. It will be seen in the next section that this is an important fact.

3.5 Controllability of Quantum Systems

We begin this section by showing that if a system is right-invariant then we need only consider the question of reachability from the identity. That is, once reachability from the identity is established, controllability follows immediately.

We can define the reachable set from any Z in $\mathcal{X}$ in terms of the set reachable from the identity. A trajectory starting from Z is given by $X(t)Z$, where $X(t)$ is a solution of the right invariant system starting at the identity. Every element $r_{\mathbb{1}}$ in the set reachable from the identity, $\mathcal{R}(\mathbb{1}_N)$, has a corresponding, though not necessarily unique element in $\mathcal{R}(Z)$ given by $r_{\mathbb{1}}Z$. The reachable set from Z is therefore

$$\mathcal{R}(Z) = \mathcal{R}(\mathbb{1}_N)Z. \tag{3.7}$$

Thus, the reachable set for any Z in $\mathcal{X}$ is completely determined by the reachable set from the identity.

We now show that the right-invariance of such a system has the additional benefit that reachability from the identity automatically implies controllability of the entire system when the state space is a Lie group.

Theorem 3.6. *Let $\mathcal{X}$ be the state space of a right invariant control system. Let $\mathcal{X}$ additionally be a Lie group with the group operation defined as X_1X_2 for any two elements X_1 and X_2 in $\mathcal{X}$ (see Appendix B for the definition of group operation.) If $\mathcal{X}$ is reachable from the identity element, $\mathbb{1}$, then the system is controllable.*

Proof. Since $\mathcal{X}$ is reachable from the identity we have that $\mathcal{R}(\mathbb{1}_N) = \mathcal{X}$. We now show that this implies $\mathcal{R}(Z) = \mathcal{X}$ for any Z in $\mathcal{X}$, and therefore that the system is controllable. Equation (3.7) states that $\mathcal{R}(Z) = \mathcal{R}(\mathbb{1}_N)Z$, so we just need to show that $\mathcal{R}(\mathbb{1}_N)Z = \mathcal{X}$.

Define the function $f(X) : \mathcal{X} \to \mathcal{X}$ by

$$f(X) = XZ,$$

Showing that $\mathcal{R}(\mathbb{1}_N)Z = \mathcal{X}$ is equivalent to showing that f maps $\mathcal{X}$ onto $\mathcal{X}$, as opposed to some proper subset of $\mathcal{X}$.

For any element X_1 in $\mathcal{X}$, there exists an element X_2 in $\mathcal{X}$ such that $f(X_2) = X_1$ given by $X_2 = X_1Z^{-1}$. The element X_2 is in $\mathcal{X}$ because both X_1 and Z^{-1} are in $\mathcal{X}$ (since every element of a group has an inverse in that group) and also because the group operation is closed.

The function f is therefore an onto mapping and it follows that $\mathcal{R}(\mathbb{1}_N)Z = \mathcal{X}$ for any Z in $\mathcal{X}$ and that the system is controllable. □

We now present and prove the controllability condition for control systems on Lie groups, which specializes to the types of quantum systems in which we are interested.

Theorem 3.7. (Controllability of Systems on Compact Lie Groups [29, Theorem 7.1]) *A right invariant control system of the form*

$$\frac{dX(t)}{dt} = -iH(v)X(t) \tag{3.8}$$

evolving on a state space $\mathcal{X}$ which is also a compact Lie group is controllable if and only if $\mathcal{X}$ is connected and the Lie algebra generated by $span_{v\in\Omega}\{-iH(v)\}$ is equal to the underlying Lie algebra, $\mathcal{L}$, of the Lie group $\mathcal{X}$.

Proof. The fact that it is necessary for $span_{v\in\Omega}\{-iH(v)\}$ to be equalt to $\mathcal{L}$ for the system to be controllable is a well-known result. See [29] for the details.

We now prove the reverse direction . By Theorem 3.6, we need only show that the reachable set from $\mathbb{1}$ is the entire state space $\mathcal{X}$. Furthermore, [29, Lemma 6.3] says that if the set reachable from the identity is dense in $\exp(\mathcal{L})$ then it is equal to $\exp(\mathcal{L})$. Thus, we need only show that $\mathcal{R}(\mathbb{1})$ is dense in $\exp(\mathcal{L})$.

Let the set of generators of $span_{v\in\Omega}\{-iH(v)\}$ be given by $L = \{L_1, L_2, \ldots, L_n\}$. When the controls are constant we can solve the system to get

$$X(t) = \exp(-iH(v)t).$$

Therefore, elements of the form $\exp(t_j L_j)$ where $t_j \geq 0$ and $L_j \in L$ are in the reachable set. By the right invariance of the system we can say that finite products of such elements are also in the reachable set.

To show that $\mathcal{R}(\mathbb{1})$ is dense in $\exp(\mathcal{L})$ we construct a sequence in $\mathcal{R}(\mathbb{1})$ which converges to elements in $\exp(\mathcal{L})$. We show that there exists a sequence of positive t_js for which

$$\lim_{j\to\infty} e^{At_j} = e^{-A|t|}.$$

Consider the sequence $e^{nA|t|}$. Since $\mathcal{X}$ is assumed to be compact, we know that this has a convergent subsequence $e^{n(j)A|t|}$. We may assume that $n(j) < n(j+1)$ for all k. This gives us that

$$e^{-A|t|} = \lim_{j\to\infty} e^{(n(j+1)-n(j)-1)A|t|}.$$

So we can construct sequences in the reachable set $\mathcal{R}(\mathbb{1})$ which converge to elements of $e^{\mathcal{L}}$. This proves that $\mathcal{R}(\mathbb{1})$ is dense in $e^{\mathcal{L}}$ which proves that the reachable set is the entire state space and controllability follows. □

Remark 3.2. The key property which allows us to show controllability is the fact that the Lie group is compact. Without compactness, controllability could not be guaranteed. The unitary group is known to be compact [17, §I, Theorem 1], so this result applies to the quantum control systems we are considering.

Remark 3.3. Theorem 3.7 is commonly referred to as the Lie algebra rank condition for controllability.

We mention a few interesting properties of systems which have Hamiltonians of the form

$$H(t) = H_{int} + H_{ext}(v).$$

In the case where there is no internal Hamiltonian (that is $H_{int} = 0$), the control system is called *homogeneous*. Such a system is significantly easier to work with in practice as it has a couple of nice properties which are pointed out in [29]. First, any point in the state space of a homogeneous system that can be reached from the identity with an unrestricted control may also be reached with a bang-bang control. Secondly, every point in the state space which can be reached from the identity can be reached in arbitrarily short time if unrestricted controls are allowed.

Neither of these results hold in general for a non-homogenous control system on a Lie group. It turns out that since $\mathcal{SU}(N)$ is compact, the first result still holds for our system (Theorem 6.5 of [29]). The second result does not hold, however, as we have seen when discussing the use of unbounded controls in [33]. From [33] we know that the final states that can be reached in arbitrarily short time with an unbounded control are the elements of the Lie subgroup associated with the Lie algebra

$$\mathcal{L}_C = \{H_1, H_2, \ldots, H_m\}_{LA}.$$

This is the Lie algebra generated by the control Hamiltonians and is called the controllability subalgebra of the system.

Required Number of Controls

In many classical control systems, controllability can be achieved with only one control even for complicated systems with many degrees of freedom. Is the same true in quantum systems?

We begin with a simple case. In [44] it is shown that any two linearly independent elements of $\mathfrak{su}(2)$ generate $\mathcal{SU}(2)$. That result, in addition to the compactness of $\mathcal{SU}(2)$ is enough to conclude the system can be controllable with as few as one control as long as its associated control Hamiltonian is linearly independent from the drift Hamiltonian. We now consider whether or not this can be generalized to higher dimensional spin systems.

A partial answer comes from [2]. Lemma 4 of this paper states that the set of pairs (A, B) in $\mathfrak{su}(N)$ which generate all of $\mathfrak{su}(N)$ is open and dense in $\mathfrak{su}(N)$. This

implies that any two elements of the Lie algebra $su(N)$ picked at random to be the internal and external Hamiltonians would give a controllable system. This, in turn, means that the system can be controllable with as few as one control.

This is not the complete answer, however, because we do not have the luxury of selecting system Hamiltonians at random. The Hamiltonians allowed are a property of the physical systems and are determined empirically. They often do not allow us to have a controllable system with only one control. In NMR quantum computing, where the free Hamiltonian is often in the form of a linear chain of coupling terms in the z axis and the control Hamiltonians selectively interact with a single qubit in either the x or y axes, it almost always requires twice the number of controls as there are qubits in the system to guarantee controllability. This certainly adds to the complexity of finding control laws.

Controllability Algorithm

An algorithm may be generated to determine whether or not the system is controllable. From [21]:

Step 1. List vectors $\{A_1, \ldots, A_s\}$ which form a basis for $\text{span}_{v\in\Omega}\{-iH(v)\}$.

Step 2. Calculate the Lie brackets of the vectors from the previous step to get the vectors for step N

Step 3. Select the vectors from this set which are linearly independent

Step 4. Return to Step 1 unless there are no new linearly independent vectors or if the dimension of the set is $N^2 - 1$ (for special unitary systems) or N^2 (for unitary systems).

The system is controllable if the final dimension of the set is $N^2 - 1$ or N^2. A Matlab implementation of this algorithm may be found in the Appendix D.

Chapter 4

The Maximum Principle

Having established criteria for the controllability of the quantum systems in question, we move on to the important task of actually controlling them. As mentioned previously, the mathematical tool we use for finding time-optimal controls is Pontryagin's Maximum Principle (PMP). PMP is also referred to as Pontryagin's Minimum Principle. The different name comes from a different application; though, the method itself is the same. This chapter discusses the principle as well as the generalization which allows us to apply it to the quantum systems of interest.

PMP is an important and often used result in optimal control theory. It gives necessary conditions for a control $v(t)$ to produce optimal trajectories. The principle originally appeared in [52] and we will follow that version of it closely in this chapter.

4.1 Pontryagin's Maximum Principle

Consider the vector differential equation

$$\dot{x}(t) = f(x, v), \tag{4.1}$$

where f is defined for $x \in \mathcal{X}$ and $v \in \Omega$. $\mathcal{X}$ and Ω are the state space and set of admissible controls respectively as introduced in the previous chapter.

Given two points x_0 and x_f in $\mathcal{X}$, we want to find an admissible control ($v \in \Omega$) which drives the system from x_0 at the initial time t_0 to x_f at the final time t_f. We want this to be done in an optimal way. To formalize what we mean by "optimal" we must introduce the idea of a cost.

Definition 4.1. *The* cost functional *is*

$$\mathcal{J} = \int_{t_0}^{t_f} L(x(t), v(t))dt. \tag{4.2}$$

The integrand $L(x, v)$ is assumed to be continuous with continuous partial derivatives on all of $\mathcal{X} \times \Omega$.

In the set of all controls which drive the system from x_0 to x_f, the optimal control is the one which minimizes the value of $\mathcal{J}$. It is possible that there is more than one control which produces the same minimum value of $\mathcal{J}$. Mathematically, all controls which drive the system from the initial to final state with the same minimum value of $\mathcal{J}$ are considered optimal. In practice, we usually only need one of them.

PMP is closely related to the method of Lagrange multipliers from optimization. It can be thought of as a generalization of this method to functionals. In the same way that the method of Lagrange multipliers introduces additional variables, PMP introduces additional state variables in the form of what is called the *costate*, denoted by $\lambda(t)$.

Definition 4.2. *The* Pontryagin Hamiltonian *(not to be confused with the Hamiltonians of quantum systems) is a function which contains both the state and the costate and is given by*

$$\mathscr{H} = \lambda_0 L(x(t), v(t)) + \langle \lambda(t), f(x(t), v(t)) \rangle ,$$

where $\langle , \rangle$ represents the standard dot product in $\mathbb{R}^n$.

The costate $\lambda(t)$ satisfies the auxiliary system

$$\frac{d\lambda_j}{dt} = -\frac{\partial \mathscr{H}}{\partial x_j} \tag{4.3}$$

for $j = 1, ..., n$. We can also recover the original system equations by

$$\frac{dx_j}{dt} = \frac{\partial \mathscr{H}}{\partial \lambda_j}, \tag{4.4}$$

for $j = 1, ..., n$.

With these definitions in mind we are now in a position to state the Maximum Principle.

Theorem 4.1. (Pontryagin's Maximum Principle [52, §2]) *Let $v(t)$, $t_0 \leq t \leq t_f$, be an admissible control such that the corresponding trajectory $x(t)$ which begins at the point x_0 at the time t_0 passes, at some time t_f, through the point x_f. In order that $v(t)$ and $x(t)$ minimize J it is necessary that there exist a constant, $\lambda_0 \leq 0$, and a continous vector function $\lambda(t) = (\lambda_1(t), ..., \lambda_n(t))$ which are not both zero, corresponding to $v(t)$ and $x(t)$, such that:*

- *Condition 1. For every $t_0 \leq t \leq t_f$, the function $\mathscr{H}(\lambda(t), x(t), v)$ of the variable $v \in \Omega$ attains its maximum at the point $v = v(t)$*

$$\mathscr{H}(\lambda(t), x(t), v(t)) = \sup_{v \in \Omega} \mathscr{H}(\lambda(t), x(t), v)$$

- *Condition 2. At the terminal time, t_f, the relation*

$$\sup_{v\in\Omega} \mathscr{H}(\lambda(t_f), x(t_f), v) = 0$$

is satisfied. Furthermore, if $\lambda(t)$, $x(t)$ and $v(t)$ satisfy (4.3), (4.4) and condition 1, the time function $\sup_{v\in\Omega}(\mathscr{H}(\lambda, x, v))$ is constant. Condition 2 may therefore be verified at any time t, $t_0 \leq t \leq t_f$.

Proof. See [52]. □

To every control input to the system, $v(t)$, there is an associated *trajectory*, $x(t)$. We refer to the pair (x, v) as a *trajectory-control pair*. The Maximum Principle states that a trajectory-control pair (x, v) can only be optimal (in other words it minimizes the cost, $\mathcal{J}$) if it maximizes the Hamiltonian, $\mathscr{H}$, for all times. This is why the result is called the Maximum Principle. A trajectory-control pair that minimizes the cost $\mathcal{J}$ is called an *extremal pair* or simply, an *extremal*.

4.2 Complex Matrix Systems

The Maximum Principle as stated in the previous section does not directly apply to the quantum systems in which we are interested. This is because such systems are complex and written in terms of matrices. Since it is not hard to write a complex matrix system as an equivalent real vector system, one can imagine that it would be possible to generalize PMP to such systems.

In [3], a maximum principle is stated for real matrix systems without proof. A different work by Baillieul [4] gives a maximum principle for minimizing energy which applies to real Lie groups and relies on Krener's higher order maximum principle, which is found in [42]. The result can be easily extended to complex systems as well. The textbook [28] also presents a generalization of the Maximum Principle to control systems on manifolds, which requires some knowledge of ideas from differential geometry. A more recent publication [24] derives a maximum principle specifically for time-optimal quantum systems, although the result is for quantum state control, not operator control.

All of [3], [4], [24] and [28] have useful insight and much of that is incorporated into this section. The remainder of this section sets up the complex matrix problem and states a version of the Maximum Principle which is appropriate for the problem we are addressing while still being general enough to be applicable to other problems.

For a complex matrix system, the dynamics are given by

$$\frac{dX(t)}{dt} = F\left(X(t), v(t)\right) \tag{4.5}$$

where X and F are elements of $\mathbb{C}^{N\times N}$ for each time t.

Even though the state variables are complex-valued, the cost functional $\mathcal{J}$ in the complex matrix case may still only take on real scalar values, otherwise it would make no sense to try to minimize it. We define the cost by

$$\mathcal{J} = \int_{t_0}^{t_f} L\left(X(t), v(t)\right) dt, \tag{4.6}$$

where $L(X(t), u(t))$ is a real-valued function which must satisfy the same continuity and differentiability requirements as the function $L(x(t), u(t))$ from the original PMP.

We define the complex matrix Pontryagin Hamiltonian as

$$\tilde{\mathscr{H}}(\lambda_0, \Lambda(t), X(t), v(t)) = \lambda_0 L + \mathrm{Re}\left\langle \Lambda(t), F(X(t), v(t))\right\rangle,$$

where $\Lambda(t)$ is an $N \times N$ costate matrix and where $<,>$ is the standard matrix inner product on $\mathbb{C}^{N\times N}$

$$\langle X, Y\rangle = \mathrm{Tr}(X^\dagger Y),$$

as given in Section 2.1 and used in [4]. This definition of the Pontryagin Hamiltonian is a straight-forward generalization of Definition 4.2 to complex systems. The following result generalizes PMP to complex matrix systems. The proof is in the next section.

Theorem 4.2. (Complex Matrix Maximum Principle) *Let $v(t)$, $t_0 \le t \le t_f$, be an admissible control such that the corresponding trajectory of $X(t)$ which begins at X_0 at time t_0 passes, at some time t_f, through the point X_f. In order that $v(t)$ and $X(t)$ minimize $\mathcal{J}$ it is necessary that there exists a real constant, $\lambda_0 \le 0$, and a continous matrix function $\Lambda(t)$ which are not both zero corresponding to $v(t)$ and $X(t)$ such that*

- *Condition 1. For every $t_0 \le t \le t_f$, the function $\tilde{\mathscr{H}}(\lambda_0, \Lambda(t), X(t), v)$ of the variable $v \in \Omega$ attains its maximum at the point $v = v(t)$*

$$\tilde{\mathscr{H}}(\Lambda(t), X(t), v(t)) = \sup_{v\in\Omega} \tilde{\mathscr{H}}(\Lambda(t), X(t), v)$$

- *Condition 2. The relation*

$$\sup_{v\in\Omega} \tilde{\mathscr{H}}(\Lambda(t), X(t), v) = 0$$

  is satisfied for all $t_0 \le t \le t_f$.

The costate matrix $\Lambda(t)$ evolves under the dynamics

$$\frac{d}{dt}\Lambda(t) = -\frac{\partial \tilde{\mathscr{H}}}{\partial X}$$

while the system state equations are recovered by

$$\frac{d}{dt}X(t) = \frac{\partial \tilde{\mathscr{H}}}{\partial \Lambda}.$$

The costate equations are given in terms of the derivitive with respect to a complex matrix. We define the derivitive of a function with respect to a complex matrix X in terms of the real and imaginary parts of X, which are X_R and X_I respectively, by

$$\frac{df(X)}{dX} = \frac{\partial f(X)}{\partial X_R} + i\frac{\partial f(X)}{\partial X_I}.$$

The derivitive of a function $f : \mathbb{R}^{N\times N} \to \mathbb{R}$ with respect to a real $N \times N$ matrix is defined component-wise in [3, Equation 15] as

$$\left(\frac{df(X)}{dX}\right)_{jk} = \frac{\partial f(X)}{\partial X_{jk}}. \tag{4.7}$$

The complex matrix version of PMP states that for a trajectory-control pair (X, v) to be extremal, it must maximize the Hamiltonian $\tilde{\mathscr{H}}$ for all times, which is the same as the result for the real vector case.

4.3 Proof of the Complex Matrix Maximum Principle

The section is devoted to proving the Complex Matrix Maximum Principle (CMMP) stated in the previous section. The details of the proof are not necessary for understanding the subsequent chapters and so may be skipped if desired. The proof is in the spirit of [3], but provides the details as well as the complex generalization. We begin with a definition and two lemmas.

Definition 4.3. *The invertible vectorization mapping,*

$$\mathrm{vec}\,(Z) : \mathbb{C}^{N\times M} \mapsto \mathbb{C}^{NM}$$

is defined by

$$\mathrm{vec}\left(\begin{bmatrix} z_1 & z_2 & \cdots & z_M \end{bmatrix}\right) = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_M \end{bmatrix},$$

where each z_j is an N-dimensional column vector. This is simply the function which stacks the columns of a matrix to produce a column vector.

Lemma 4.1. *Let P and Q be complex $N \times N$ matrices such that*

$$\begin{aligned} P &= P_R + iP_I \\ Q &= Q_R + iQ_I. \end{aligned}$$

Let the corresponding real $2N^2$-dimensional vectors p and q be defined by

$$p = \mathrm{vec}\left(\begin{bmatrix} P_R & P_I \end{bmatrix}\right)$$

and

$$q = \mathrm{vec}\left(\begin{bmatrix} Q_R & Q_I \end{bmatrix}\right)$$

The following statement holds

$$\mathrm{Re}\,\langle P, Q\rangle_{\mathbb{C}^{N\times N}} = \langle p, q\rangle_{\mathbb{R}^{2N^2}}\,.$$

Proof. See Appendix A. □

Lemma 4.2. *For any matrix-valued function which takes a matrix as an argument, $G(X(t))$, there exists a matrix-valued function, $\tilde{G}(x(t))$, which takes $x(t) = \mathrm{vec}(X(t))$ as an argument and is equal to $G(X(t))$ for all time.*

Proof. We prove the existence of such a function by construction.

Since $\mathrm{vec}(X(t))$ is invertible, we can say $\mathrm{vec}^{-1}(x(t)) = X(t)$. So we have that

$$\begin{aligned} G(X(t)) &= G(\mathrm{vec}^{-1}(x(t))) \\ &= \tilde{G}(x(t)), \end{aligned}$$

which completes the proof. □

Theorem 4.2. (The Complex Matrix Maximum Principle)

Proof. The proof proceeds by mapping the complex matrix system into an equivalent real vector system, applying PMP and then mapping the results back onto the complex matrix space. Throughout the proof we will omit the arguments of the functions unless a step requires that they be shown explicitly.

The first step of the proof is to rewrite the system (4.5) as an equivalent real system of the form

$$\dot{x} = f(x, v), \tag{4.8}$$

We begin by breaking the system into real and imaginary parts

$$\dot{X}_R + i\dot{X}_I = F_R + iF_I.$$

This expression is equivalent to the real $N \times 2N$ matrix system given by

$$\begin{bmatrix} \dot{X}_R & \dot{X}_I \end{bmatrix} = \begin{bmatrix} F_R & F_I \end{bmatrix}.$$

Using Lemma 2, we replace $F_R(X, v)$ and $F_I(X, v)$ with equivalent functions in terms of $x = \mathrm{vec}\,(X)$,

$$\begin{bmatrix} \dot{X}_R & \dot{X}_I \end{bmatrix} = \begin{bmatrix} \tilde{F}_R(x, v) & \tilde{F}_I(x, v) \end{bmatrix}$$

We vectorize both sides to get

$$\dot{x} = f(x, v),$$

a real vector system where $f(x,v) = \text{vec}\left(\begin{bmatrix} \tilde{F}_R(x,v) & \tilde{F}_I(x,v) \end{bmatrix}\right)$. We also use Lemma 4.2 to find an equivalent cost function for the real vector system

$$\tilde{L}(x, v) = L(X, v).$$

$\tilde{L}$ must satisfy the the assumptions of PMP since it is equal to L which satisfies the same assumptions. We have now written (4.5) in a form to which we may apply PMP.

The second step of the proof is to apply PMP to this system and then to map the results into the complex matrix form. By PMP we know that for $x(t)$ and $v(t)$ to be optimal there must exist λ_0 and $\lambda(t)$ which are not both 0 such that the expression for $\mathscr{H}$ given by

$$\mathscr{H}(\lambda, \lambda_0, x, v) = \lambda_0 L_R(x, v) + \langle \lambda, f(x, v) \rangle,$$

is maximized and equal to zero for all time. Using the definitions of $\tilde{L}$ and f, we rewrite $\mathscr{H}$ as

$$\begin{aligned} \mathscr{H}(\lambda, \lambda_0, x, v) &= \lambda_0 L(X, v) + \left\langle \lambda, \text{vec}\left(\begin{bmatrix} \tilde{F}_R(x,v) & \tilde{F}_I(x,v) \end{bmatrix}\right) \right\rangle \\ &= \lambda_0 L(X, v) + \left\langle \lambda, \text{vec}\left(\begin{bmatrix} F_R(X,v) & F_I(X,v) \end{bmatrix}\right) \right\rangle, \end{aligned}$$

where the second line follows from Lemma 4.2.

Finally, we define the function $\Lambda(t)$ as $\Lambda(t) = \Lambda_R(t) + i\Lambda_I(t)$, and in turn define $\Lambda_R(t)$ and $\Lambda_I(t)$ in terms of $\lambda(t)$ by

$$\begin{bmatrix} \Lambda_R & \Lambda_I \end{bmatrix} = \text{vec}^{-1}_{N \times 2N}(\lambda).$$

Inverting and substituting into our expression for $\mathscr{H}$,

$$\begin{aligned} \mathscr{H}(\lambda, \lambda_0, x, v) &= \lambda_0 L + \left\langle \text{vec}\left(\begin{bmatrix} \Lambda_R & \Lambda_I \end{bmatrix}\right), \text{vec}\left(\begin{bmatrix} F_R & F_I \end{bmatrix}\right) \right\rangle_{\mathbb{R}^{2N}} \\ &= \lambda_0 L + \text{Re}\left\langle \Lambda(t), F(X, v) \right\rangle_{\mathbb{C}^{N \times N}} \\ &= \tilde{\mathscr{H}}(\Lambda, \lambda_0, X, v), \end{aligned}$$

where the second line follows from Lemma 4.1. Note that the complex matrix Pontryagin Hamiltonian is exactly equal to the Pontryagin Hamiltonian of the equivalent real vector system. This means that any result that holds for $\mathscr{H}$ also holds for $\tilde{\mathscr{H}}$. Note also that the existence of $\lambda(t)$ in the original maximum principle implies the existence of $\Lambda(t)$ in this generalization and that the constant λ_0 is the same for both Hamiltonians.

It remains to show how to determine the complex matrix form of the costate dynamics. This can be done by applying the mappings we used to transform our system into real vector equations in reverse. From PMP we know that

$$\frac{d\lambda}{dt} = \begin{bmatrix} \frac{d\lambda_1}{dt} \\ \vdots \\ \frac{d\lambda_{2N^2}}{dt} \end{bmatrix} = - \begin{bmatrix} \frac{\partial \mathscr{H}}{\partial x_1} \\ \vdots \\ \frac{\partial \mathscr{H}}{\partial x_{2N^2}} \end{bmatrix}$$

We apply the inverse of the vec function and use the definitions of Λ_R and Λ_I to get

$$\mathrm{vec}^{-1}_{N\times 2N}\left(\frac{d\lambda}{dt}\right) = \frac{d}{dt}\left(\left[\begin{array}{cc}\Lambda_R & \Lambda_I\end{array}\right]\right) = -\left[\begin{array}{ccc}\frac{\partial \mathscr{H}}{\partial x_1} & \cdots & \frac{\partial \mathscr{H}}{\partial x_{(2N^2-N+1)}} \\ \cdots & \ddots & \cdots \\ \frac{\partial \mathscr{H}}{\partial x_N} & \cdots & \frac{\partial \mathscr{H}}{\partial x_{2N^2}}\end{array}\right].$$

We note that although the vector has been reshaped, the equation is otherwise unchanged. Using the component-wise definition of derivative we can rewrite this expression more compactly as

$$\left[\begin{array}{cc}\frac{d\Lambda_R}{dt} & \frac{d\Lambda_I}{dt}\end{array}\right] = -\left[\begin{array}{cc}\frac{\partial \mathscr{H}}{\partial X_R} & \frac{\partial \mathscr{H}}{\partial I_r}\end{array}\right].$$

This can in turn be expressed as a single complex $N \times N$ matrix equation by rewriting it as the equivalent complex system

$$\frac{d}{dt}(\Lambda_R + i\Lambda_I) = -\left(\frac{\partial \tilde{\mathscr{H}}}{\partial X_R} + i\frac{\partial \tilde{\mathscr{H}}}{\partial X_I}\right),$$

which is the same as

$$\frac{d}{dt}\Lambda(t) = -\frac{\partial \tilde{\mathscr{H}}}{\partial X}.$$

An identical calculation gives

$$\frac{d}{dt}X(t) = \frac{\partial \tilde{\mathscr{H}}}{\partial \Lambda}, \tag{4.9}$$

which completes the proof. □

As a check, we note that the original system equations are indeed recovered by (4.9) since

$$\begin{aligned}\frac{d}{dt}X(t) &= \frac{\partial \tilde{\mathscr{H}}}{\partial \Lambda_R} + i\frac{\partial \tilde{\mathscr{H}}}{\partial \Lambda_I} \\ &= \left(\frac{\partial}{\partial \Lambda_R} + i\frac{\partial}{\partial \Lambda_I}\right)(\Lambda_0 L + \mathrm{Re}\,\langle \Lambda, F\rangle) \\ &= \frac{\partial}{\partial \Lambda_R}(\mathrm{Re}\,\langle \Lambda, F\rangle) + i\frac{\partial}{\partial \Lambda_I}(\mathrm{Re}\,\langle \Lambda, F\rangle).\end{aligned}$$

The expression $\mathrm{Re}\,\langle \Lambda, F\rangle$ can be simplified by breaking F and Λ into the real and imaginary parts $F = F_R + iF_I$ and $\Lambda = \Lambda_R + i\Lambda_I$. Therefore,

$$\begin{aligned}\mathrm{Re}\,\langle \Lambda, F\rangle &= \mathrm{Re}\,\langle \Lambda_R + i\Lambda_I, F_R + iF_I\rangle \\ &= \mathrm{Re}\,(\langle \Lambda_R, F_R\rangle - i\,\langle \Lambda_I, F_R\rangle + i\,\langle \Lambda_R, F_I\rangle + \langle \Lambda_I, F_I\rangle) \\ &= \langle \Lambda_R, F_R\rangle + \langle \Lambda_I, F_I\rangle.\end{aligned}$$

We now use Theorem A.3 (Appendix A) which states that

$$\frac{d}{dX}\langle X, A\rangle = A$$

and we obtain

$$\begin{aligned}\frac{d}{dt}X(t) &= \frac{\partial}{\partial \Lambda_R}\langle \Lambda_R, F_R\rangle + i\frac{\partial}{\partial \Lambda_I}\langle \Lambda_I, F_I\rangle \\ &= F_R + iF_I \\ &= F,\end{aligned}$$

which are the original system dynamics, as expected.

4.4 Specialization To Unitary Evolution

We now apply the complex matrix maximum principle to the quantum system discussed in Chapter 2 to find necessary conditions on the control input $v(t)$ to drive the system to a desired unitary operator in the minumum time.

Recall that the quantum system dynamics are given by the Schrödinger operator equation

$$\dot{X}(t) = -i\left(H_0 + \sum_{j=1}^{m} v_j(t)H_j\right)X(t). \tag{4.10}$$

To find time-optimal solutions, we must choose $L(X(t), v(t)) = 1$. The cost functional is therefore given by

$$\mathcal{J} = \int_{t_0}^{t_f} L(X(t), v(t))dt = \int_{t_0}^{t_f} 1dt.$$

From this definition we see that the cost $\mathcal{J}$ is equal to the total time. Minimizing $\mathcal{J}$ therefore minimizes the total time, as desired. Without loss of generality, we set $t_0 = 0$ for the remainder of this report.

With the system dynamics given by (4.10), the complex matrix Pontryagin Hamiltonian, $\mathscr{H}$, is given by

$$\mathscr{H} = \lambda_0 + \mathrm{Re}\,\langle \Lambda(t), -iH(v)X(t)\rangle\,. \tag{4.11}$$

where, for simplicity, we have written $H(v) = H_0 + \sum_{j=1}^{m} v_j(t)H_j$.

From the CMMP, the costate equation is given by

$$\frac{d\Lambda(t)}{dt} = -\left(\frac{\partial \mathscr{H}}{\partial X_R} + i\frac{\partial \mathscr{H}}{\partial X_I}\right).$$

Theorem 4.3. *The costate evolves under the same dynamics as the state, that is*

$$\frac{d\Lambda(t)}{dt} = -iH(v)\Lambda(t), \tag{4.12}$$

though the initial condition Λ_0 may be different from X_0.

Proof. The proof is an algebraic excerice. Refer to Appendix C for the details. □

From Theorem 2.2 we know that (4.10) is a right invariant system. This allows us to simplify the expression for the Hamiltonian. A similar result appears in [4].

Theorem 4.4. *The Hamiltonian*

$$\mathscr{H} = \lambda_0 + \operatorname{Re}\langle \Lambda(t), -iH(v)X(t)\rangle$$

is equivalent to

$$\mathscr{H} = \lambda_0 + \operatorname{Re}\left\langle M, -iX(t)^\dagger H(v)X(t)\right\rangle,$$

where M is a constant $N\times N$ matrix. In other words, the Hamiltonian is completely determined by X, λ_0 and a constant matrix M.

Proof. Since (4.10) is a right invariant system, any solution $X(t)$ of (4.10) multiplied on the right by an arbitrary matrix M is also a solution.

We have already shown that $X(t)$ and $\Lambda(t)$ evolve under the same dynamics. If we let $M = \Lambda(0)$, then we have that $X(0)M = \Lambda(0)$, since $X(0) = \mathbb{1}$.

We know that both $\Lambda(t)$ and $X(t)M$ are solutions of (4.10). Since they also have the same initial condition, by uniqueness ([29, Lemma 2.1]) it follows that

$$\Lambda(t) = X(t)M.$$

We can therefore rewrite the Hamiltonian completely in terms of X, M and λ_0 as

$$\mathscr{H} = \lambda_0 + \operatorname{Re}\langle X(t)M, -iH(v)X(t)\rangle.$$

By the definition of the inner product, we have that

$$\begin{aligned}\mathscr{H} &= \lambda_0 + \operatorname{Re}\left\{\operatorname{Tr}((X(t)M)^\dagger(-iH(v)X(t)))\right\}\\ &= \lambda_0 + \operatorname{Re}\left\{\operatorname{Tr}(M^\dagger X^\dagger(t)(-iH(v)X(t)))\right\}\\ &= \lambda_0 + \operatorname{Re}\left\langle M, -iX^\dagger(t)H(v)X(t)\right\rangle,\end{aligned}$$

as required. □

We have that

$$\mathscr{H} = \lambda_0 + \operatorname{Re}\left\langle M, -iX(t)^\dagger H(v)X(t)\right\rangle,$$

where λ_0 and M cannot simultaneously be zero (by the CMMP).

Theorem 4.5. *The matrix M from the Pontryagin Hamiltonian*

$$\mathscr{H} = \lambda_0 + \operatorname{Re}\left\langle M, -iX^\dagger(t)H_jX(t)\right\rangle$$

is an element of $\mathfrak{su}(N)$.

Proof. See Appendix B □

Theorem 4.6. *The trace inner product between two elements of $\mathfrak{u}(n)$ is always real.*

Proof. Let A and B be elements of $\mathfrak{u}(n)$, that is A and B are skew-Hermitian matrices. The trace inner product is given by

$$\begin{aligned}\langle A, B\rangle &= \operatorname{Tr}(A^\dagger B)\\ &= \operatorname{Tr}(-AB)\\ &= -\operatorname{Tr}(BA)\\ &= -\left\langle B^\dagger, A\right\rangle\\ &= \langle B, A\rangle\\ &= \overline{\langle A, B\rangle}.\end{aligned}$$

This equality holds if and only if $\langle A, B\rangle$ is real. □

It is easy to verify that $-iX^\dagger(t)H_jX(t)$ is a skew-Hermitian matrix for each H_j. Therefore by Theorems 4.5 and 4.6 we can write the Pontryagin Hamiltonian as

$$\mathscr{H} = \lambda_0 + \left\langle M, -iX(t)^\dagger H(v)X(t)\right\rangle,$$

where the matrix M is an element of $\mathfrak{su}(N)$.

Using the definition of $H(t)$ we can expand this as

$$\mathscr{H} = \lambda_0 + \left\langle M, -iX(t)^\dagger H_0X(t)\right\rangle + \sum_{j=1}^{m} v_j(t)\left\langle M, -iX(t)^\dagger H_jX(t)\right\rangle. \tag{4.13}$$

We further simplify the notation by defining

$$\psi_j(t) = \left\langle M, -iX(t)^\dagger H_jX(t)\right\rangle. \tag{4.14}$$

The ψ_js are referred to as the *switching functions*. This terminology will be clear by the end of the chapter. The Hamiltonian now has the form

$$\mathscr{H} = \lambda_0 + \psi_0(t) + \sum_{j=1}^{m} v_j(t)\psi_j(t). \tag{4.15}$$

4.5 Types of Extremals

We are considering a system of the form

$$\dot{X}(t) = -iH\left(v(t)\right)X(t), \quad X(0) = \mathbb{1},$$

where the components of v, $v_k(t) \in [0, v_{max}]$. We want to drive this system to some specified final unitary state X_d in the minimum possible time. From the previous section we see that applying the Maximum Principle to the above systems gives a Pontryagin Hamiltonian of the form

$$\mathscr{H}_P = \psi_0(t) + \sum_{k=1}^{m} \psi_k(t) v_k(t) + \lambda_0,$$

where

$$\psi_j(t) = \left\langle M, X^\dagger(t)(-iH_j)X(t) \right\rangle, \quad j = 0, 1, \ldots, m$$

for some constant, traceless skew-Hermitian matrix, M. Both M and λ_0 cannot be simultaneously zero.

Normal and Abnormal Extremals

It is possible for λ_0 to be identically zero. Extremals for which $\lambda_0 \equiv 0$ are called *abnormal*. All other extremals are called *normal*. These two cases are usually treated separately and could give different control laws. In many simple problems, it can be shown that abnormal extremals cannot be optimal. In the quantum state control problem addressed in [5], it was shown that abnormal extremals could not exist. On the other hand, [20] characterizes both normal and abnormal extremals for the time-optimal control of a two qubit system. The general case of the unitary operator synthesis problem remains an open question.

Using the definitions of the switching functions in terms of inner products, we can write the Hamiltonian as

$$\mathscr{H} = \lambda_0 + \left\langle M, -iX^\dagger(t)H(t)X(t) \right\rangle,$$

which must be equal to zero. If an extremal is abnormal, we have that

$$\left\langle M, -iX^\dagger(t)H(t)X(t) \right\rangle \equiv 0$$

or, geometrically speaking, that M is orthogonal to $-iX^\dagger(t)H(t)X(t)$ for all time. This idea could help answer the question of whether or not abnormal extremals may exist.

Singular Extremals

The CMMP states that it is necessary for an extremal to maximize $\mathscr{H}$ over the admissible controls. In this system's Hamiltonian, we see that the controls appear linearly. It is well known that the maximum of a linear function on a closed interval occurs on the endpoints of the interval. As a result we get that

$$v_j(t) = \begin{cases} v_{max}, & \text{if } \psi_j(t) > 0 \\ 0, & \text{if } \psi_j(t) < 0 \end{cases},$$

since $v_k(t)$ is restricted to

$$v_k(t) \in [0, v_{max}].$$

Notice that the control only takes on the highest or lowest possible value allowed and switches between the two when the switching function $\psi_j(t)$ changes from being positive to negative. Such a control law is referred to as a *bang-bang* control.

However, in the case where a switching function is identically zero over an open time interval, the above control law does not apply. Extremals which have some open interval on which a switching function is identically zero are called *singular*. All other extremals are called *non-singular*. In simple problems, the switching functions can sometimes be computed explicitly. Determining whether singular extremals are possible in such a case is a simple matter of looking at the form of the switching functions. It is not always possible to do this in the general case.

For an optimal control problem with bang-bang controls to be completely solved we must treat the case of singular extremals. This can be done in one of three ways. The first way is to prove that singular extremals cannot exist. The second is to prove that if singular extremals do exist, that they cannot be optimal and, therefore, need not be considered. The final way is to find an optimal control law which holds on intervals where the switching function is identically zero.

A good reference for dealing with singular extremals is [27]. In it, the authors provide an approach to determining whether or not it is possible to have singlular extremals in a particular problem. A Pontryagin Hamiltonian of the form

$$\mathscr{H} = I(t) + F(t)v(t)$$

is considered, where $I(t)$ and $F(t)$ are functions that come out of the construction of $\mathscr{H}$. The Hamiltonian in our problem is the multiple control generalization of this Hamiltonian. If $F(t)$ is identically zero on an open interval, then this is a singular interval. We also know from PMP that optimal extremals require that $\mathscr{H} \equiv 0$ for all times, so we have the additional conditions on singular intervals that

$$I(t) = \dot{I}(t) = \ddot{I}(t) = \cdots \equiv 0$$

and

$$F(t) = \dot{F}(t) = \ddot{F}(t) = \cdots \equiv 0.$$

If these conditions cannot be satisfied on any interval then there can be no singular extremals. In the quantum control problem we have multiple controls. To consider singular intervals for the j^{th} control, the function $F(t)$ is given by $\psi_j(t)$ and the function $I(t)$ is given by

$$I(t) = \psi_0(t) + \sum_{k \neq j} \psi_k(t) v_k(t) + \lambda_0.$$

The function $\dot{F}(t)$ can be calculated explicitly in our problem by

$$\begin{aligned}
\frac{dF(t)}{dt} &\equiv \frac{d}{dt} \left\langle M, -iX^\dagger(t) H_k X(t) \right\rangle \\
&\equiv \frac{d}{dt} \mathrm{Tr} \left(M^\dagger (-iX^\dagger(t) H_k X(t)) \right) \\
&\equiv -i \mathrm{Tr} \left(M^\dagger \frac{dX^\dagger(t)}{dt} H_k X(t) + M^\dagger X^\dagger(t) H_k \frac{dX(t)}{dt} \right) \\
&\equiv -i \mathrm{Tr} \left(M^\dagger (-iHX)^\dagger H_k X + M^\dagger X^\dagger H_k (-iHX) \right) \\
&\equiv \mathrm{Tr} \left(M^\dagger X^\dagger (H H_k - H_k H) X \right) \\
&\equiv \left\langle M, X^\dagger \left[H, H_k \right] X \right\rangle .
\end{aligned}$$

So we have that

$$\dot{F}(t) = \dot{\psi}_k(t) = \left\langle M, X^\dagger(t) \left[H(v), H_k \right] X(t) \right\rangle . \tag{4.16}$$

It is difficult to determine whether or not this will be identically zero on an interval.

We note that the conditions we stated for $F(t)$, $I(t)$ and their derivitives are necessary conditions for singular extremals to exist. Even if they are satified singular extremals may still not occur. Section 4.6 examines this question more closely in the case of a single qubit with two controls.

4.6 One Qubit Example

The unitary evolution of a system of one qubit under the influence of an external magnetic field is given by the time-dependent Schrödinger operator equation

$$\dot{X}(t) = -iH(t)X(t), \quad X(0) = \mathbb{1}_2, \tag{4.17}$$

where $X(t)$ is a 2×2 unitary matrix and $H(t)$ is a Hermitian matrix given by

$$H(t) = H_0 + v_1(t) H_1 + v_2(t) H_2. \tag{4.18}$$

The function $v_j(t)$ is the control input corresponding to the j^{th} control Hamiltonian and may represent the amplitude of radio field of a particular frequency. The Hamiltonians are given by

$$\begin{aligned}
H_0 &= 2\pi I_z \\
H_1 &= 2\pi I_x \\
H_2 &= 2\pi I_y.
\end{aligned} \tag{4.19}$$

This is a one qubit NMR quantum computer setup. The internal Hamiltonian H_0 corresponds to the coupling of the qubit to the background magnetic field. This field induces a rotation of the spin state around the z axis. The external Hamiltonians H_1 and H_2 correspond to the influence of magnetic fields which induce rotation of the spin state around the x and y axis, respectively. The amplitudes of these fields are the control inputs to the system.

The I_α matrices are the Pauli spin matrices given by

$$I_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad I_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad I_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

In addition to the property that $I_\alpha^2 = \mathbb{1}$ for $\alpha = x$, y and z, the Pauli matrices satisfy

$$I_x I_y = iI_z, \quad I_y I_z = iI_x, \quad I_z I_x = iI_y \tag{4.20}$$

where reversing the order of multiplication changes the result by a negative sign. It is well-known that the Pauli matrices generate $\mathfrak{su}(2)$. The system is therefore controllable by the Lie algebra rank condition introduced in Chapter 3.

The references [33] and [68] consider time-optimal control of the same system, except where $H_2 = 0$; that is, they consider the one control version of the same system. We also use a different admissible control set than either of those papers. In [33], an unbounded control is used while in [68] a control with the restriction

$$|v(t)| \leq B,$$

is used. This lower bound on the magnitude of this control is $-B$ and not zero as in the problem we are considering in this section. The authors of [68] state without proof that time-optimal controls must either be full strength ($u_k(t) = \pm B$) or zero at each time t. This is almost a bang-bang control law except that the control must be zero on a singular interval. The authors of [33] find the minimum time and show that the control takes the form of a strong pulse, followed by an interval where the system is allowed to drift, followed by another strong pulse.

The Pontryagin Hamiltonian for the control system defined by (4.17-4.19) is

$$\mathscr{H}(t) = \psi_0(t) + \psi_1(t)v_1(t) + \psi_2(t)v_2(t) + \lambda_0.$$

As before, the switching functions, $\psi_j(t)$ are given by

$$\psi_j(t) = \left\langle M, -iX^\dagger(t)H_j(v)X(t) \right\rangle$$

for some constant, traceless, skew-Hermitian matrix M. As in the previous section, the maximization of the Hamiltonian implies a bang-bang control law in the absence of singular intervals.

Switching Functions

We wish to determine whether or not singular intervals or abnormal extremals may exist for this problem. To do so, we will examine the switching functions more closely. We first note that the switching functions are not independent of each other. Referring to equation (4.16),

$$\dot{\psi}_j(t) = \left\langle M, X^\dagger(t)\left[H(t), H_j\right]X(t)\right\rangle.$$

From the definition of $H(t)$, and the properties of inner products and commutators we obtain

$$\begin{aligned}\dot{\psi}_j(t) &= \left\langle M, X^\dagger(t)\left[H_0, H_j\right]X(t)\right\rangle \\ &+ \left\langle M, X^\dagger(t)\left[H_1, H_j\right]X(t)\right\rangle v_1(t) \\ &+ \left\langle M, X^\dagger(t)\left[H_2, H_j\right]X(t)\right\rangle v_2(t).\end{aligned}$$

We now use the Pauli matrix relations from equation (4.20) to simplify the expression:

$$\begin{aligned}\dot{\psi}_0(t) &= 4\pi\left(v_2(t)\left\langle M, -iX^\dagger(t)H_1X(t)\right\rangle - v_1(t)\left\langle M, -iX^\dagger(t)H_2X(t)\right\rangle\right) \\ \dot{\psi}_1(t) &= -4\pi\left(v_2(t)\left\langle M, -iX^\dagger(t)H_0X(t)\right\rangle + \left\langle M, -iX^\dagger(t)H_2X(t)\right\rangle\right) \\ \dot{\psi}_2(t) &= 4\pi\left(v_1(t)\left\langle M, -iX^\dagger(t)H_0X(t)\right\rangle - \left\langle M, -iX^\dagger(t)H_1X(t)\right\rangle\right).\end{aligned}$$

We can rewrite these equations completely in terms of the switching functions to obtain the system of equations

$$\begin{aligned}\dot{\psi}_0(t) &= 4\pi v_2(t)\psi_1(t) - 4\pi v_1(t)\psi_2(t) \\ \dot{\psi}_1(t) &= -4\pi v_2(t)\psi_0(t) + 4\pi\psi_2(t) \\ \dot{\psi}_2(t) &= 4\pi v_1(t)\psi_0(t) - 4\pi\psi_1(t).\end{aligned}$$

Defining

$$\Psi(t) = \left[\begin{array}{ccc}\psi_1(t) & \psi_2(t) & \psi_3(t)\end{array}\right]^T$$

and

$$Q = \left[\begin{array}{ccc}0 & v_2(t) & -v_1(t) \\ -v_2(t) & 0 & 1 \\ v_1(t) & -1 & 0\end{array}\right]$$

we may write the system more compactly as

$$\dot{\Psi}(t) = 4\pi Q\Psi(t). \tag{4.21}$$

Singular Extremals

Let us assume that we have a singular interval for the first control $v_1(t)$, that is, $\psi_1(t) \equiv 0$ for for all t in some interval $S_1 = [t_1, t_2]$. From (4.21) we obtain

$$\begin{aligned} \dot{\psi}_0(t) &= -4\pi v_1(t)\psi_2(t) && (4.22)\\ 0 &= v_2(t)\psi_0(t) - \psi_2(t) && (4.23)\\ \dot{\psi}_2(t) &= 4\pi v_1(t)\psi_0(t), && (4.24) \end{aligned}$$

for all t in S_1.

By CMMP, an optimal extremal must satisfy the additional condition

$$0 \equiv \psi_0(t) + \psi_2(t)v_2(t) + \lambda_0 \tag{4.25}$$

which is just the condition that the Pontryagin Hamiltonian must be zero for all time, including on the interval S_1.

It is possible that there are open subintervals of S_1 on which the second control is singular as well. We denote the union of these intervals by $\hat{S}_1$. We therefore have that on the intervals of $\hat{S}_1$ both $\psi_1(t)$ and $\psi_2(t)$ are identically zero. On the the intervals of $(S_1 \backslash \hat{S}_1)$, $\psi_1(t)$ is identically zero and $\psi_2(t)$ is almost everywhere non-zero. We consider the two cases separately.

In $\hat{S}_1$, we use (4.25) to solve for the switching functions and obtain

$$\begin{aligned} \psi_0(t) &= -\lambda_0 && (4.26)\\ \psi_1(t) &= 0 && (4.27)\\ \psi_2(t) &= 0. && (4.28) \end{aligned}$$

Equations (4.24), (4.26) and (4.28) yield

$$v_1(t)\lambda_0 = 0.$$

On the other hand, when $t \in (S_1 \backslash \hat{S}_1)$ then, after some algebra we obtain the switching functions

$$\begin{aligned} \psi_0(t) &= \frac{-\lambda_0}{1+v_2^2} && (4.29)\\ \psi_1(t) &= 0 && (4.30)\\ \psi_2(t) &= \frac{-v_2\lambda_0}{1+v_2^2}, && (4.31) \end{aligned}$$

which, again, imply that

$$v_1(t)\lambda_0 = 0.$$

If the extremal is normal, then $\lambda_0 \neq 0$ and this implies that $v_1(t) = 0$ on the singular interval S_1.

We can perform the same analysis for the second control as well. Let S_2 be an interval on which $\psi_2(t) \equiv 0$. A similar calculation yields the same result in terms of the second switching function,

$$v_2(t)\lambda_0 = 0.$$

Again, in the case of a normal extremal, this implies that $v_2(t) = 0$.

Notice that on a singular interval all of the switching functions (4.26-4.28) and (4.29-4.31) are constant. Since the controls only switch when a switching function changes sign it follows that once a control becomes singular no further switches may occur.

Abnormal Extremals

We showed that a singular interval for the j^{th} control implies that

$$v_j(t)\lambda_0 = 0$$

for all t on that interval. The remaining problem is that if $\lambda_0 = 0$ then we cannot determine $v_j(t) \equiv 0$.

If the extremals are abnormal, then by setting λ_0 to zero in equations (4.26-4.31) we obtain

$$\psi_j(t) = 0$$

for $j = 0, 1$ and 2 on any singular interval.

By the definition of the switching functions

$$\psi_j(t) = \left\langle M, -iX^\dagger(t)H_j(t)X(t) \right\rangle,$$

so M must be orthogonal to $-iX^\dagger(t)H_j(t)X(t)$ for $j = 0, 1$ and 2.

In the case of a one-qubit system, $X^\dagger(t)(-iH_j)X(t)$ forms a basis for $\mathfrak{su}(2)$ for all t, since $X(t)$ is unitary and since $-iH_j$ forms a basis for $\mathfrak{su}(2)$. This means that M is orthogonal to every element of a basis for $\mathfrak{su}(2)$. The only element of $\mathfrak{su}(2)$ which is simultaneously orthogonal to every element of a basis is the zero element. This violates the assumptions of the Maximum Principle, namely that λ_0 and M cannot both simultaneously be zero so such a trajectory cannot be optimal and need not be considered.

We have not proved that abnormal extremals do not exist for the one qubit case, but that singular intervals in conjunction with abnormal extremals cannot be optimal. For a normal extremal, a singular interval for a particular control implies that the associated control must be identically zero on that interval. We therefore have the bang-bang control law

$$v_j(t) = \begin{cases} v_{max}, & \text{if } \psi_j(t) > 0 \\ 0, & \text{if } \psi_j(t) \leq 0 \end{cases}.$$

If we did not have $v_{min} = 0$, then $v_j(t)$ could take the values v_{min}, 0 or v_{max}.

Number of Switches

We have shown that on any interval between switching times the control must be constant. This means that we can solve the system (4.21). The solution is given by

$$\begin{aligned}\psi_1(t) &= \alpha_1 \sin(\omega t + \phi_1) - \frac{v_1}{K}\lambda_0 \\ \psi_2(t) &= \alpha_2 \sin(\omega t + \phi_2) - \frac{v_2}{K}\lambda_0,\end{aligned}$$

where

$$\begin{aligned}\alpha_1 &= \frac{1}{K}\sqrt{\frac{K}{J}v_2^2A^2 + K^2B^2 + JKC^2 + 2K(v_1\lambda_0 B - v_2 AC) + v_1^2\lambda_0^2} \\ \alpha_2 &= \frac{1}{K}\sqrt{\frac{K}{J}v_1^2A^2 + KJB^2 + K^2C^2 + 2K(v_2\lambda_0 C - v_1 AB) + v_2^2\lambda_0^2},\end{aligned}$$

$$\begin{aligned}\phi_1 &= \arctan(\sqrt{\frac{J}{K}}\gamma_1) + \begin{cases} 0, & \gamma_1 \geq 0 \\ \pi, & \gamma_1 < 0 \end{cases} \\ \phi_2 &= \arctan(\sqrt{\frac{J}{K}}\gamma_2) + \begin{cases} 0, & \gamma_2 \geq 0 \\ \pi, & \gamma_2 < 0 \end{cases},\end{aligned}$$

and where

$$\begin{aligned}\gamma_1 &= \frac{\lambda_0 v_1 + BK}{v_2 A + JC} \\ \gamma_1 &= \frac{\lambda_0 v_2 + CK}{JB - v_1 A}.\end{aligned}$$

The scalars A, B and C correspond to the initial condition

$$\Psi_0 = \begin{bmatrix} A \\ B \\ C \end{bmatrix},$$

and the period $\omega = 4\pi\sqrt{J(J + v_1^2 + v_2^2)}$.

Having an explicit description of the switching functions allows us to gain some insight into what time-optimal controls will look like for this problem. Specifically, it is clear that there cannot be an infinite number of switches in any finite interval of time since these are just sine functions plus a constant. In a controllable system such as this one, all final states are reachable in finite time and so there must be a finite number of switches.

We can also see that the frequency of switches is related to the period ω. If $|\alpha_i| > |\frac{v_i}{K}\lambda_0|$, then $\psi_i(t)$ will switch signs eventually. The period of the switching function $\frac{2\pi}{\omega}$ is an upper bound on the time until the next switch.

4.7 Systems with More Than One Qubit

It is clearly of interest to know how much of the analysis used for the single qubit case can be carried over to the general case. Unfortunately, a straightforward generalization of these results has not been found. Furthermore, even the prospects for finding such a generalization are not promising. The reason for this is that writing dynamical system equations for the switching functions becomes much more difficult with additional qubits.

As a concrete example to illustrate the problems, consider the case of a two qubit NMR implementation of quantum computing which is considered in [33]. The quantum Hamiltonian of the system has the form

$$H(t) = H_0 + \sum_{j=1}^{4} H_j v_j(t)$$

where

$$\begin{aligned} H_0 &= 2\pi J(I_z \otimes I_z) \\ H_1 &= 2\pi(I_x \otimes \mathbb{1}) \\ H_2 &= 2\pi(I_y \otimes \mathbb{1}) \\ H_3 &= 2\pi(\mathbb{1} \otimes I_x) \\ H_4 &= 2\pi(\mathbb{1} \otimes I_y) \end{aligned}$$

and J is a constant term which indicates the strength of the coupling between the two qubits.

The Pontryagin Hamiltonian for the system has the form

$$\mathscr{H} = \lambda_0 + \psi_0(t) + \sum_{j=1}^{4} H_j v_j(t),$$

where the switching functions are

$$\psi_j(t) = \left\langle M, -iX^\dagger(t) H_j X(t) \right\rangle.$$

The derivatives of the switching functions are

$$\dot{\psi}_j(t) = \left\langle M, X^\dagger(t)[H, H_j]X(t) \right\rangle.$$

The problem arises when evaluating the Lie brackets $[H, H_j]$. In the one qubit case, these brackets could be expressed in terms of the other H_j's, which would allow the switching functions to be written in terms of other switching functions. We are able to do this for a system of one qubit because the dimension of $\mathfrak{su}(2)$ is three.

The set $\{-iH_j\}$ for $j = 0, 1$ and 2 is linearly independent and therefore forms a basis for $\mathfrak{su}(2)$.

In the two qubit case, the bracket $[H, H_j]$ cannot be expressed in terms of the H_j's anymore because the dimension of $\mathfrak{su}(4)$ is fifteen and therefore the set $\{-iH_j\}$ for $j = 0, \ldots, 4$ cannot possibly span the entire space. For a system of q qubits, the Hamiltonians are elements of $\mathfrak{su}(2^q)$, which has dimension $(2^q)^2 - 1$. We will only have at most $2q$ control Hamiltonians in addition to the drift Hamiltonian. As a result, it is unlikely that we could write the Lie bracket $[H, H_j]$ as a linear combination of the elements of $\{-iH_j\}$ except in the one qubit case. The best we could hope for would be to take higher order derivitives of the switching functions and to write the resulting nested Lie brackets as functions of the Hamiltonians and the Lie brackets of the Hamiltonians.

Even if it is possible to write out a dynamical system for the switching functions using higher order derivitives, it is not unreasonable to expect that the dynamical system of switching functions would have the same dimension as the underlying physical system, as it does for the one qubit system. If this is the case, then the dimension of the system of switching functions would increase exponentially in the number of qubits, making already tedious calculations all but intractable.

Adding to the complication is that the form of the quantum system Hamiltonian depends on the physical implementation of the quantum computer. Any dynamical system of switching functions would depend heavily on this and so would need to be computed for every different system.

While there is still the promise of further analytical advance on this problem, the approach used in the one qubit case does not seem practical for larger systems. For that reason, in the next chapter we turn our attention to numerical solutions of the time-optimal quantum control problem.

Chapter 5

Computing Optimal Controls

The remainder of this thesis considers the computation of (hopefully) time-optimal controls. Specifically, we devise an algorithm which finds bang-bang controls for the quantum system we have discussed. Recall from Chapter 4.4 that in the absence of singular intervals the optimal control law will be bang-bang. If there are singular intervals, then a bang-bang control may not be time-optimal but will still be able to steer the system from the identity to the correct final state, as proved in Chapter 3.

In the case of the one qubit system discussed in Section 4.6, the optimal control law is bang-bang since on singular intervals the control is zero, which is the lower bound on the control inputs. It is hoped that this is also true in higher-dimensional cases or, at least, that the effect of singular intervals might be small so that bang-bang controls will be very close to time-optimal.

5.1 Existing Numerical Methods

Before outlining the method, we take a closer look at existing numerical approaches to controlling quantum systems. The first approach we discuss is not so much a numerical method as a heuristic approach [46]. The method, called insensitive nuclei enhanced by polarization transfer (INEPT) was introduced in 1978 and is used to enhance the intensity of NMR signals for certain types of nuclei. The pulse sequence that comes from INEPT can also be viewed as synthesizing logic gates [62, §2.1]. As such, it was initially used to develop control strategies for NMR implementations of quantum computing.

More recently, new methods have been discovered based on standard control and optimization principles. For example, the relaxed optimized pulse element (ROPE) algorithm introduced in [39] uses results from dynamical programming to provide a more efficient pulse sequence than INEPT in many cases.

Subsequent work by the first author of [39] resulted in the gradient ascent pulse engineering (GRAPE) [38, 58], which has gained much attention for the purpose

of synthesizing unitary operators. The GRAPE algorithm fixes a final time T and discretizes the evolution into time-steps of equal width, Δt. A gradient descent algorithm is then used to maximize the function

$$\Phi = |\langle X_d, X_f(T)\rangle|^2,$$

where X_d is the desired final unitary operator and $X_f(T)$ is the unitary operator actually generated at time T. The global maximum of Φ occurs when $X_d = X_f(T)$. The optimization variables are the values of the controls v_k on each time step. Running the algorithm with smaller and smaller values of T until the final desired state cannot be reached is a way to find a time-optimal control.

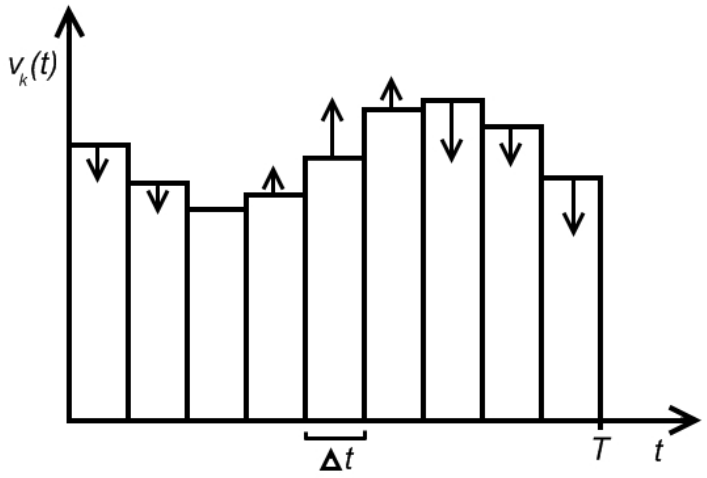


Figure 5.1: An iteration of the gradient ascent pulse engineering method. The arrows represent the gradient, which is calculated at each iteration to minimize Φ. The final time T is fixed.

Figure 5.1 shows graphically how GRAPE calculates optimal controls. The small arrows represent the direction of the gradient at each time-step. This is the direction in which to update the control on that interval to achieve a final unitary operator which is closer to the desired one.

In practice, it is usually much more computationally efficient to use a conjugate gradient method as opposed to the simpler gradient ascent method. Such a method has been implemented at the University of Waterloo's Institute for Quantum Computing for computing controls for their NMR experiments [54]. For an introduction to the conjugate gradient method, refer to [59].

5.2 Computational Framework

We wish to solve the problem of time-optimal control of the system

$$\dot{X} = -iH(v)X. \tag{5.1}$$

We initially set up the framework for a control system with a single control. The algorithm is generalized to multiple controls in Section 5.4. Since we are only considering bang-bang controls, the solution is completely determined by the initial control $v(0)$ and the knowledge of when the control switches. The times when the control switches are called *switching times*. In terms of the Maximum Principle, these correspond to the times when a switching function is zero, with the exception of $\psi_0(t)$. Let there be $k+1$ switching times denoted by τ_j for $j = 0, 1, \ldots k$. We define τ_0 to be the initial time zero, and τ_k to be the final time t_f. We write the set of switching times as the vector

$$\tau = (\tau_0, \ldots, \tau_k)^T.$$

On any open interval (τ_{j-1}, τ_j), the control input is constant and denoted by v^j. Since the time dependence of $H(v)$ enters only through the control, $H(v^j)$ is constant on that interval as well. For simplicity, we define $H^j := H(v^j)$. It is also useful to define the vector

$$\xi = (\xi_1, \xi_2, \ldots, \xi_k)^T,$$

the vector of interval lengths where

$$\xi_j = \tau_j - \tau_{j-1}.$$

The final state of system (5.1) with bang-bang controls is therefore given by the matrix product

$$X_f(\xi) = \exp(-iH^k\xi_k) \ldots \exp(-iH^2\xi_2) \exp(-iH^1\xi_1) X_0,$$

where X_0 is the initial state.

Since we want to reach a particular final state, we introduce a measure of how close we are to that state. The function $\Phi(\xi)$ is defined as

$$\Phi(\xi) = \frac{1}{2} \langle X_f(\xi) - X_d, X_f(\xi) - X_d \rangle ,$$

where X_d is the desired final unitary operator. This function gives a measure of the distance between the final state of our system for a given set of switching times and the desired final state . By the properties of inner products, $\Phi(\xi) \geq 0$ for all time with equality if and only if $X_f(\xi) = X_d$. Minimizing $\Phi(\xi)$ is therefore equivalent to finding a vector of interval lengths associated to a set of switching times which drives the system from the initial state to the desired final state.

The following result is useful for simplifying $\Phi(\xi)$.

Theorem 5.1. *Let X be an arbitrary element of $\mathcal{U}(N)$. Then*

$$\langle X, X \rangle = N$$

Proof.

$$\begin{aligned}\langle X, X\rangle &= \mathrm{Tr}(X^\dagger X)\\ &= \mathrm{Tr}(\mathbb{1}_N)\\ &= N\end{aligned}$$

□

Using the properties of inner products and Theorem 5.1 we obtain

$$\begin{aligned}\Phi(\xi) &= \frac{1}{2}\left(\langle X_f(\xi), X_f(\xi)\rangle - \langle X_f(\xi), X_d\rangle - \langle X_d, X_f(\xi)\rangle + \langle X_d, X_d\rangle\right)\\ &= \frac{1}{2}\left(2N - \langle X_f(\xi), X_d\rangle - \overline{\langle X_f(\xi), X_d\rangle}\right)\\ &= \frac{1}{2}\left(2N - 2\mathrm{Re}\,\langle X_f(\xi), X_d\rangle\right)\\ &= N - \mathrm{Re}\,\langle X_f(\xi), X_d\rangle\,.\end{aligned}$$

For the purposes of most optimization algorithms it is important to know the gradient as well as the Hessian of $\Phi(\xi)$. The gradient is

$$\begin{aligned}\frac{\partial\Phi(\xi)}{\partial\xi_q} &= -\frac{\partial}{\partial\xi_q}\left(\mathrm{Re}\,\langle X_f(\xi), X_d\rangle\right)\\ &= -\frac{\partial}{\partial\xi_q}\left(\mathrm{Re}\left(\mathrm{Tr}\left(X_f^\dagger(\xi)X_d\right)\right)\right)\\ &= -\mathrm{Re}\left(\mathrm{Tr}\left(\frac{\partial X_f^\dagger(\xi)}{\partial\xi_q}X_d\right)\right)\\ &= -\mathrm{Re}\left(\mathrm{Tr}\left(\frac{\partial X_f(\xi)^\dagger}{\partial\xi_q}X_d\right)\right)\\ &= -\mathrm{Re}\left\langle\frac{\partial X_f(\xi)}{\partial\xi_q}, X_d\right\rangle.\end{aligned}$$

From the definition of X_f, we calculate $\frac{\partial X_f(\xi)}{\partial\xi_q}$ to be

$$\frac{\partial X_f(\xi)}{\partial\xi_q} = \left(\prod_{j=k}^{q+1}\exp(-iH^j\xi_j)\right)(-iH^q)\left(\prod_{j=q}^{1}\exp(-iH^j\xi_j)\right)X_0,$$

where we note that the indices are decreasing. This is to reinforce that the order of the multiplication should be taken to be in order of decreasing indices from left to right.

We may calculate the second partial derivatives to be used in the Hessian in a similar manner. We find that

$$\begin{aligned}\frac{\partial^2\Phi(\xi)}{\partial\xi_p\partial\xi_q} &= -\frac{\partial}{\partial\xi_p}\mathrm{Re}\left\langle\frac{\partial X_f(\xi)}{\partial\xi_q}, X_d\right\rangle\\ &= -\mathrm{Re}\left\langle\frac{\partial^2 X_f(\xi)}{\partial\xi_p\xi_q}, X_d\right\rangle.\end{aligned}$$

When $p \geq q$, defining $E_j = \exp(-iH^j)$ allows us to write the second derivatives as

$$\frac{\partial^2 X_f(\xi)}{\partial \xi_p \partial \xi_q} = -\left(\prod_{j=k}^{p+1} E_j \xi_j\right)(H^p)\left(\prod_{j=p}^{q+1} E_j \xi_j\right)(H^q)\left(\prod_{j=q}^{1} E_j \xi_j\right) X_0,$$

using the same decreasing index convention as before. Interchanging p and q on the right side of the equation gives the expression in the case where $p < q$.

5.3 Time-Optimal Switching Algorithm

The algorithm we use to compute the time-optimal bang-bang controls is based on the time-optimal switching (TOS) algorithm developed by Kaya et al in [30, 31, 32, 45] and which is outlined in this section.

The first step of the algorithm is to compute a feasible, but not necessarily optimal, bang-bang control which takes the system from the initial to the final point. The aforementioned references have called the algorithm used to complete this step the switching time control (STC) algorithm.

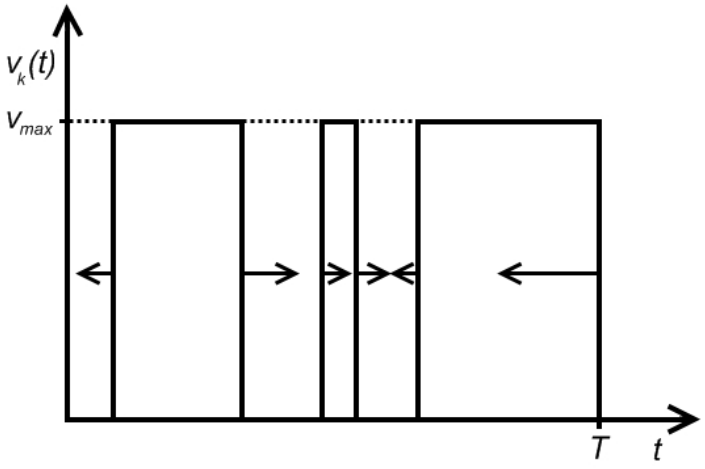


Figure 5.2: Switching time control iteration. The arrows represent the gradient, which is used to update the switching times at each iteration. The final time T is not fixed, but the magnitude of the control must be either zero or v_{max}.

The STC algorithm starts with an initial guess ξ_g for the switching times and then minimizes the function $\Phi(\xi)$ using a suitable optimization algorithm (such as the conjugate gradient method, for example.) Note that if there are not enough intervals in ξ_g it may not be possible to find a point such that $\Phi(\xi) = 0$. When this occurs it is necessary to add more intervals to the initial guess until a solution of $\Phi(\xi) = 0$ can be found. Figure 5.2 shows graphically how an iteration of the

STC algorithm proceeds. The arrows indicate the gradients used to update the switching times at each iteration.

Note the difference between this approach and the approach of GRAPE from Figure 5.1. Specifically, while GRAPE fixes a final time, the final time in the STC algorithm is free to vary. Instead, the STC algorithm fixes the possible values of the control function so that the control law is bang-bang. The STC implementation used for the results obtained in this report uses a conjugate gradient method to find feasible solutions.

We consider the set of points ξ for which $X_f(\xi) = X_d$ to be a surface in ξ-space as follows. We define the function

$$S(\xi) := X_f(\xi) - X_d.$$

The set of points where $S(\xi) = \mathbb{0}$ gives the desired surface. We refer to this as the *terminal surface*. In words, this is the set of possible interval values ξ which drive the quantum system from the identity to the desired final unitary operator. Using this terminology we say that the STC algorithm finds some point on the terminal surface. Note that the terminal surface is also equivalent to the level set $\Phi(\xi) = 0$.

Having found a point on the terminal surface, ξ_0, the second step of the algorithm is to move around on this surface so as to find the point which minimizes the final time

$$t_f(\xi) = \sum_{j=1}^{k} \xi_j = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} \xi.$$

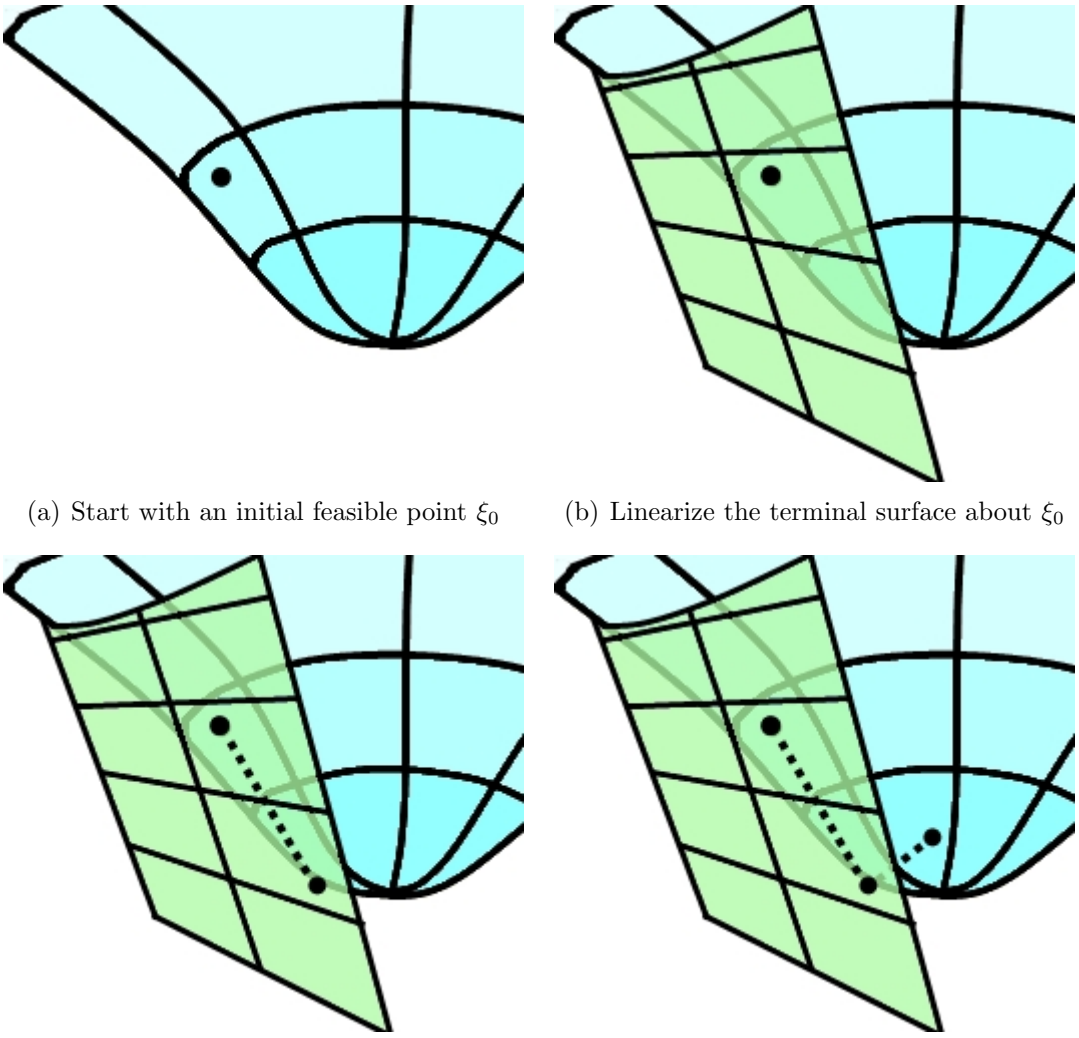
This is a nonlinear programming problem where the nonlinearity enters through the constraint that ξ be restricted to the terminal surface.

To solve this problem, we linearize the terminal surface S about the feasible point ξ_0 using the standard formula for a tangent surface

$$S_L(\xi) = \nabla S(\xi_0) \cdot (\xi - \xi_0) = 0.$$

This set of linear constraints in addition to the linear function $t_f(\xi)$ that we wish to minimize and the restriction that the each interval length be non-negative, $\xi_k \geq 0$, is in the standard form of a Linear Programming (LP) problem. The standard algorithms for this problem are either the simplex method or an interior point method. Denote the solution of the LP problem by ξ_{LP}. We refer to the move from the initial feasible point ξ_0 to ξ_{LP} as a full step in the direction of ξ_{LP}.

For the implementation in this thesis, we use Matlab's built-in `linprog` command which uses an interior point method for large-scale problems and a simplex method variant for smaller problems. We then project from the solution of the LP problem back onto the terminal surface since this point, which was found from the linearization, will not satisfy the terminal constraints in general. Figure 5.3 gives a visual description of how step 2 of the method proceeds in the case of a 2-dimensional ξ-space.



(a) Start with an initial feasible point ξ_0

(b) Linearize the terminal surface about ξ_0

(c) Solve the linear programming problem

(d) Project back to the terminal surface

Figure 5.3: Step 2 of time optimal switching algorithm.

Note that the projection back to the terminal surface is a non-trivial step. In Kaya's algorithm a quasi-Newton method is used. This implementation yielded poorly conditioned matrices for our problem. As a result, we reuse the STC algorithm for this job. An algorithm which searches in a direction orthogonal to the linearized terminal surface might produce better results but has not yet been implemented.

It is not always possible to project back to the terminal surface. When the projection back to the terminal surface fails, we halve the step size by choosing the point

$$\xi_{half} = \frac{\xi_{LP} + \xi_0}{2}.$$

We continue to halve the step size until it is possible to return to the terminal surface.

Once a minimum of $\Phi(\xi)$ has been reached there are a few cases to consider. The first possibility is that all of the intervals have a strictly positive length. In this case it may be possible that the initial number of intervals was insufficient to achieve the optimal result. We therefore append a zero length interval to the end of the set of intervals and set the value of the control on this interval to zero if the previous control is v_{max} and to v_{max} if the previous control is zero. We also add another interval to the start of the set in the same way. We then use this as the initial feasible point and repeat the linearization and projection step.

The other possibility is that the lengths of some of the intervals have become zero. We say that an interval with zero length has *collapsed*. These intervals have no effect on the final state of the system and are removed. After removing collapsed intervals, some consecutive intervals may have the same value of the control. These intervals are merged together to form a single interval. Since intervals have collapsed, adding more intervals is not likely to provide any benefit and the algorithm terminates. To summarize, the TOS algorithm proceeds as follows.

Step 1. Calculate a feasible solution, ξ_0, with the STC algorithm.

Step 2. (a) Linearize the terminal surface about ξ_0 and solve the LP problem.

(b) Project back to the terminal surface. If it is not possible to return to the terminal surface or if the new point on the terminal surface does not reduce the final time then halve the step size until a point on the terminal surface with a smaller final time can be found. Set ξ_0 to be this new point.

Step 3. Repeat Step 2 until the solution of the linear programming problem is ξ_0.

Step 4. If no intervals have collapsed then add a zero length interval before the the first interval as well as an additional zero length interval after the last arc and return to Step 2. If at least one interval has collapsed proceed to Step 5.

Step 5. Remove collapsed intervals and merge arcs which have common controls being applied if necessary.

5.4 Multiple Controls

We now move on to consider a system with multiple controls. Control problems with an arbitrary number of controls were not considered in [30, 31, 32, 45]. The presence of additional control functions complicates the algorithm, though the basic idea is the same. For the remainder of this chapter we assume that the system has m controls.

Recall that for the single control algorithm the k-dimensional vector ξ contains the lengths of intervals between switching times. For multiple controls we define the vector ξ in the same way, that is, it is the vector of interval lengths between

any two consecutive switching times. Note that the consecutive switching times are not necessarily associated to the same control. For that reason we must also keep track of which control is switching at each switching time. We therefore define a vector $\mathcal{C}$ in $\mathbb{R}^k$ whose j^{th} element is the number of the control that switches at the j^{th} switching time τ_j.

It is also useful to keep track of the interval times between switches of the same control. For this purpose we define a vector of interval times for each of the m controls, denoted α_j where j is the index of the control to which the vector is associated. We have that

$$\alpha_j = \begin{bmatrix} \alpha_{j1} \\ \vdots \\ \alpha_{j\kappa_j} \end{bmatrix}, \quad j = 1, \ldots, m$$

where α_{jl} is the interval length between the l^{th} and $(l-1)^{\text{st}}$ switching times associated to the j^{th} control. Since the number of intervals need not be the same for each control we define the vector κ in $\mathbb{R}^m$ with components $(\kappa_1, \ldots, \kappa_m)$ which stores the number of intervals associated to each control. It follows that each α_j is a κ_j-dimensional vector. In practice it is easiest to implement all of the α_j vectors as columns in a single matrix, where the columns with fewer entries are filled with placeholder zeros. We denote this matrix by α and refer to it as the *interval matrix*. The interval matrix α has the form

$$\alpha = \begin{bmatrix} \alpha_1^* & \alpha_2^* & \ldots & \alpha_m^* \end{bmatrix},$$

where

$$\alpha_j^* = \begin{bmatrix} \alpha_j \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

In the single control case, the matrix α has just one column that coincides with ξ. Similarly, the vector κ reduces to a scalar that coincides with k.

The matrix of interval lengths associated to each control α and the vector of overall interval lengths between any two controls ξ are both useful in the multiple control TOS (MCTOS) algorithm. As such, it is often necessary to convert between the two viewpoints. It is possible to define an invertible linear transformation B between the vector ξ and the matrix α such that

$$\xi = B(\alpha, \mathcal{C})$$

and

$$\alpha = B^{-1}(\xi, \mathcal{C}),$$

where $\mathcal{C}$ is the vector that contains the order of switchings as defined previously. Knowledge of the vector $\mathcal{C}$ is necessary to perform the transformation. For more information on how this is done, see Appendix E.

The multiple control version of the STC algorithm is nearly identical to the single control version. An initial guess is selected and expressed in terms of ξ, the vector of intervals between switching times for the overall system. We also need a guess of which controls are being switched at each switching time so that we may determine the value of the controls on each interval. Once these are selected, the STC algorithm finds an initial feasible point. The order of which control switches at each switching time is not changed in this initial step.

For the second step, which involves repeatedly linearizing and projecting back onto the terminal surface, it is more convenient to work with the interval matrix α. This is because we want each interval to be non-negative. Enforcing this condition for the vector ξ would lead to inequality constraints, which we choose to avoid. Additionally, we rewrite the interval matrix by stacking its columns since standard optimization algorithms take vectors as arguments as opposed to matrices. We define the vector σ by

$$\sigma = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix},$$

which conveniently excludes the placeholder zeros from the matrix α. Excluding the placeholder zeros requires knowledge of the number of intervals associated to each control, which is stored in the vector κ (and may also be computed from $\mathcal{C}$). We allow the invertible transformation B from above to act on σ as well as α since they both contain the same elements in a different arrangement. It is technically necessary to include this in the transformation B but we omit this detail for simplicity. Likewise, we say that

$$\sigma = B^{-1}(\xi, \mathcal{C}).$$

Whether $B^{-1}(\xi, \mathcal{C})$ gives α or σ should be inferred from the context.

The optimization problem is now to minimize

$$t_f(\sigma) = \sum_{j=1}^{\kappa_1} \sigma_j = \begin{bmatrix} 1 & \dots & 1 & 0 & \dots & 0 \end{bmatrix} \sigma,$$

subject to the constraint that

$$S(B(\sigma, \mathcal{C})) = X_f(B(\sigma, \mathcal{C})) - X_d = 0$$

and where $\sigma_j \geq 0$. We also require that each control stops at the same time. This gives the additional equality constraints

$$\sum_{j=1}^{\kappa_1} \alpha_{1j} = \sum_{j=1}^{\kappa_2} \alpha_{2j} = \cdots = \sum_{j=1}^{\kappa_m} \alpha_{mj}.$$

We have stated the constraint in terms of α for notational simplicity and note that it is a simple matter to rewrite this in terms of σ. For a system with m controls, there will be an additional $m-1$ constraints of this form.

This is almost the same nonlinear programming problem as for the single control case with the exception of the additional constraints and the objective function $t_f(\sigma)$. In the single control case, the objective function can be found by simply adding up all of the intervals. In the multiple control case we only add up the intervals associated to one of the controls. By convention we always use the first control for this purpose in this thesis. Note that the terminal function takes $B(\sigma, \mathcal{C})$ as its argument, which is the same as the single control case since $B(\sigma, \mathcal{C}) = \xi$.

If we linearize the constraints arising from the terminal condition, we again get a linear programming problem in standard form. This is solved exactly the same way as for one control before projecting back to the terminal surface and repeating. Note that after solving the LP problem, it is possible that the order of control switches may have changed. The order of switching can also change when the step size determined by the LP problem needs to be halved. When the order of switches changes it is necessary to update the vector $\mathcal{C}$.

Once a minimum has been found, we check to see if any of the intervals have collapsed in the α representation. If any intervals have collapsed for a given control, then we remove them and merge remaining intervals if neceesary. When no intervals have collapsed for a given control, we append new zero length intervals at the start and end for that control as in the single control case and update the vector $\mathcal{C}$ accordingly. If intervals have collapsed for every control then the optimization terminates, otherwise we repeat the linearize/project steps until this happens.

The steps of the multiple control algorithm are essentially the same as the single control algorithm with the exceptions of the generalizations mentioned in this section. The MCTOS algorithm is a generalization of the TOS algorithm to an arbitrary number of controls in that the MCTOS algorithm is completely equivalent to the TOS algorithm when applied to a system with one control.

5.5 Special Considerations for Quantum Systems

One feature of the quantum system (5.1) which is not necessarily present in a general nonlinear control problem is the periodicity of solutions. For the one qubit case, the quantum Hamiltonians for each interval $\mathbb{H}_j$ are 2×2 traceless Hermitian matrices. This implies that $i\mathbb{H}_j$ is a traceless skew-Hermitian matrix or, that $i\mathbb{H}_j$ is an element of $\mathfrak{su}(2)$. Using the definition of $\mathfrak{su}(n)$ from Theorem 3.3, it is easy to show that any element A in $\mathfrak{su}(2)$ has the form

$$A = \begin{bmatrix} ia & -b+ic \\ b+ic & -ia \end{bmatrix},$$

where a, b and c are real numbers. The eigenvalues of A are

$$\mathrm{eig}(A) = \{\ i\sqrt{a^2+b^2+c^2} \quad -i\sqrt{a^2+b^2+c^2}\ \}.$$

These eigenvalues are strictly imaginary and both have the same magnitude.

This implies that the evolution of the one qubit system is periodic between switching times, although the period, henceforth denoted p_j, may be different for each such interval. This means that, in the one qubit case, the length of any interval ξ_j can always be translated to lie between zero and p_j. This is very useful for simplifying the vector of interval lengths. Unfortunately, the same result doesn't hold in general for systems with multiple qubits.

For a multiple qubit system, the state space is $\mathcal{SU}(N)$ where $N = 2^q$ and q is the number of qubits in the system. The quantum Hamiltonians for each interval $i\mathbb{H}_j$ are elements of $\mathfrak{su}(N)$. It is easy to show that the eigenvalues of these matrices are purely imaginary. Despite this, we may still not have periodic solutions because purely imaginary eigenvalues are only a necessary condition for periodic orbits. In the single qubit case, the fact that both eigenvalues must have the same magnitude allows the solutions to be periodic. In general, it is possible for magnitudes of the eigenvalues to be irrational multiples of each other which would prevent periodic solutions from occuring.

Even when the eigenvalues are rational multiples of each other, choosing the length of the interval modulo the period is not always the best approach. With multiple eigenvalues there is often a great deal of symmetry which may be exploited to reduce the lengths of the intervals. Finding an algorithm to do this efficiently remains an open problem.

Chapter 6

Simulation Results

We now apply the optimization algorithm presented in Chapter 5 to the quantum system

$$\dot{X} = -iH(v)X. \tag{6.1}$$

We consider the one qubit system discussed in Chapter 4.4 given by

$$\dot{X}(t) = -iH(v)X(t), \quad X(0) = \mathbb{1}_2,$$

where $X(t)$ is a 2×2 unitary matrix and $H(v)$ is a Hermitian matrix given by

$$H(t) = H_0 + v_1(t)H_1 + v_2(t)H_2.$$

The Hamiltonian matrices are

$$\begin{aligned} H_0 &= 2\pi J I_z \\ H_1 &= 2\pi I_x \\ H_2 &= 2\pi I_y. \end{aligned}$$

The control functions v_1 and v_2 are piecewise constant functions bounded below by zero and above by v_{max}. For the following simulations we use the values $J = 1$ and $v_{max} = 1$. In this case it is impossible to reasonably neglect the drift term because the control fields cannot be made large relative to the effect of the coupling between the qubits, represented by H_0.

6.1 One Qubit Hadamard Gate

We choose the final unitary operator

$$X_d = \frac{i}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

This is the one qubit Hadamard gate which has been multiplied by a phase factor so that it has unit determinant (and is therefore an element of $\mathcal{SU}(2)$.) In decimal notation, the Hadamard gate is

$$\text{Hadamard} = \begin{bmatrix} 0.7071i & 0.7071i \\ 0.7071i & 0.7071i \end{bmatrix}.$$

Since we are using a conjugate gradient algorithm, we expect that the solution we would get for any particular run would depend on the initial guess of the intervals ξ. This algorithm also only finds local minima. For these reasons, we run the algorithm multiple times and pick the best result from the different attempts.

Tables 6.1, 6.2 and 6.3 show the outputs for three different runs rounded to four decimal places. All three attempts are very close to the Hadamard gate. By the distance measure $\Phi(\xi)$ defined in Chapter 5, all three are within 10^{-7} of the target. By visual inspection of the final operator we can confirm that these final matrices are indeed very close to the Hadamard gate.

Intervals:

j	ξ	v_1	v_2
1	0.5303	1	0

Final Values:

$$t_f(\xi) = 0.5305$$

$$\Phi(\xi) = 3.7709 \cdot 10^{-8}$$

$$X_f(\xi) = \begin{bmatrix} 0.7070i & 0.7072i \\ 0.7072i & 0.7070i \end{bmatrix}$$

Table 6.1: Hadamard Gate with one Interval

Each of the three runs shown has a different number of intervals and corresponds to a different local minumum. We know that these are local minima because the value $\nabla\Phi(\xi)^T\nabla\Phi(\xi)$ (the squared magnitude of the gradient) is also less than 10^{-7} and the Hessian of $\Phi(\xi)$ is positive definite at each of the points. The frequency of observing each of these solutions varies with the initial guess of ξ. It has been noted that starting each control with seven intervals for each control seems to result in a higher likelihood of getting the two interval solution from Table 6.2.

Of the three runs shown, there are two solutions with the final time 0.4655. The units have not been included since the time has been rescaled by $\hbar$ as in Section 2.6. As of yet, the algorithm has not produced a solution which takes less time than the two from Tables 6.2 and 6.3, though those two solutions have reocurred frequently. It seems likely, therefore, that these are the time-optimal solutions of 7.1 for driving the system from the identity to the one qubit Hadamard gate.

Table 6.4 shows the initial feasible solution calculated by the STC algorithm at iteration 0, as well as the values of ξ after each linearization and projection back

Intervals:

j	ξ	v_1	v_2
1	0.1767	1	1
2	0.2887	1	0

Final Values:

$$t_f(\xi) = 0.4655$$

$$\Phi(\xi) = 2.6418e \cdot 10^{-8}$$

$$X_f(\xi) = \begin{bmatrix} -0.0001 + 0.7070i & 0.7072i \\ 0.7072i & -0.0001 - 0.7070i \end{bmatrix}$$

Table 6.2: Hadamard Gate with two Intervals

Intervals:

j	ξ	v_1	v_2
1	0.1936	1	1
2	0.1744	0	1
3	0.0974	1	1

Final Values:

$$t_f(\xi) = 0.4655$$

$$\Phi(\xi) = 3.2673 \cdot 10^{-8}$$

$$X_f(\xi) = \begin{bmatrix} 0.7071i & 0.0002 + 0.7072i \\ -0.0002 + 0.7072i & -0.7071i \end{bmatrix}$$

Table 6.3: Hadamard Gate with three Intervals

to the terminal surface which occurs during Step 2 of the MCTOS algorithm. This table is from the same run as Table 6.2.

In this case it took three iterations of the second step to reach the time-optimal solution. On the third iteration, all but three of the original intervals were excluded and set to zero. Note that the final two-interval form of the solution comes from merging the third and fifth intervals since they have the same values of controls.

We note that the fact that there are more than one solution with the same optimal time is interesting. It indicates, as expected, that the geometry of the optimization space is complicated. In particular, it is easy to see that the optimization space is non-convex. If the optimization space were convex, then the solution set would also have to be convex. This would mean that any point on the line connecting the two optimal solutions from Tables 6.2 and 6.3 would have to be an optimal solution as well. This is not the case because we can write the two optimal solutions as

ξ^1	ξ^2	v_1	v_2
0.1767	0.1936	1	1
0.2887	0	1	0
0	0.1744	0	1
0	0.0974	1	1

	Iterations				Controls	
j	0	1	2	3	v_1	v_2
1	0.2377	0.2200	0.4030	0	1	1
2	0.1272	0.2227	0.0007	0.1767	1	0
3	0.1545	0.0128	0.1944	0.0407	1	1
4	0.1049	0.2041	0.1882	0	0	1
5	0.1952	0.1092	0.0779	0.2480	1	1
6	0.1164	0.1778	0.1111	0	0	1
7	0.1385	0.1055	0.0910	0	1	1
8	0.0488	0.0124	0.0150	0	1	0
9	0.1132	0.0739	0.0241	0	1	1

Table 6.4: Iterations of the linearize/projection step.

The midpoint of the two solutions $\frac{\xi^1+\xi^2}{2}$ does not drive the system to the correct final state and so by counter-example the optimization space is not convex.

6.2 The Limit of Unbounded Controls

If we increase the allowed magnitude of the controls, we expect the final time required to produce a given unitary operator should decrease. In the limit as the controls magnitudes tend to infinity, we expect the solution from the MCSTC algorithm to approach solutions found using the assumption of unbounded controls.

Recall from Section 3.5 that the infimum time required to produce a unitary operator with unbounded controls is zero only if the unitary operator is in the subgroup associated to the controllability subalgebra. For the one qubit, two control example from the previous section, the controllability subalgebra is given by

$$\mathcal{L}_C = \{2\pi I_x, 2\pi I_y\}_{LA}.$$

It is trivial to show that I_x and I_y generate $\mathfrak{su}(2)$. The associated subgroup is therefore the entire state space $\mathcal{SU}(2)$. As a result, we expect that as the upper bound on the controls tends towards infinity, the minimum possible time required to produce any special unitary operator tends to zero. Figure 6.1 plots the best times found using the MCTOS algorithms against the upper bound of the control functions. We see that the final time does decrease at every step and appears to be approaching zero.

In the case where we use only one control, say v_1 or v_2, the controllability subalgebra is no longer $\mathcal{SU}(2)$. We therefore expect that the final time will not tend to zero in the limit of unbounded controls. Figure 6.2 gives the best computed minimum times when we use only the first control, v_1. Indeed, the plot indicates that the final time tends to a value greater than zero as expected.

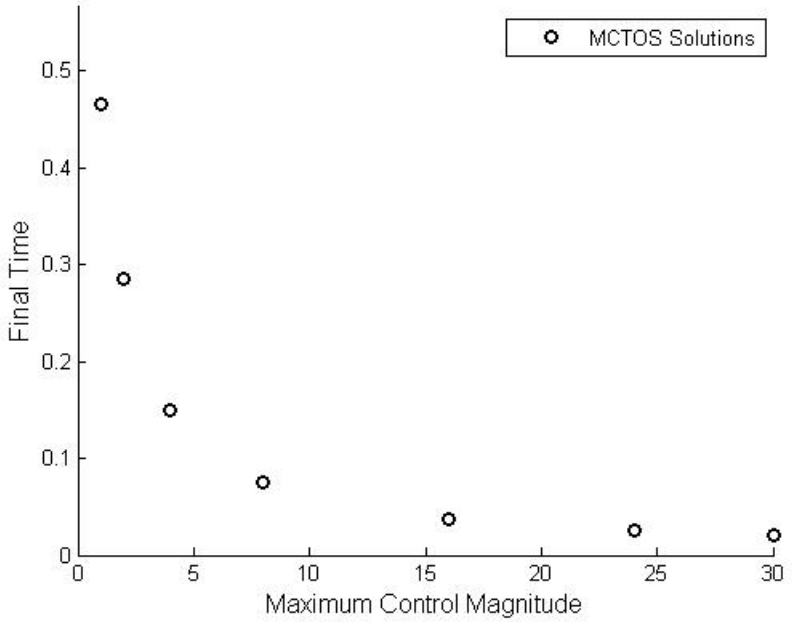


Figure 6.1: Final time vs. maximum control amplitude. The circles give the minimum time required to compute the one qubit Hadamard gate with a given upper bound on the control input.

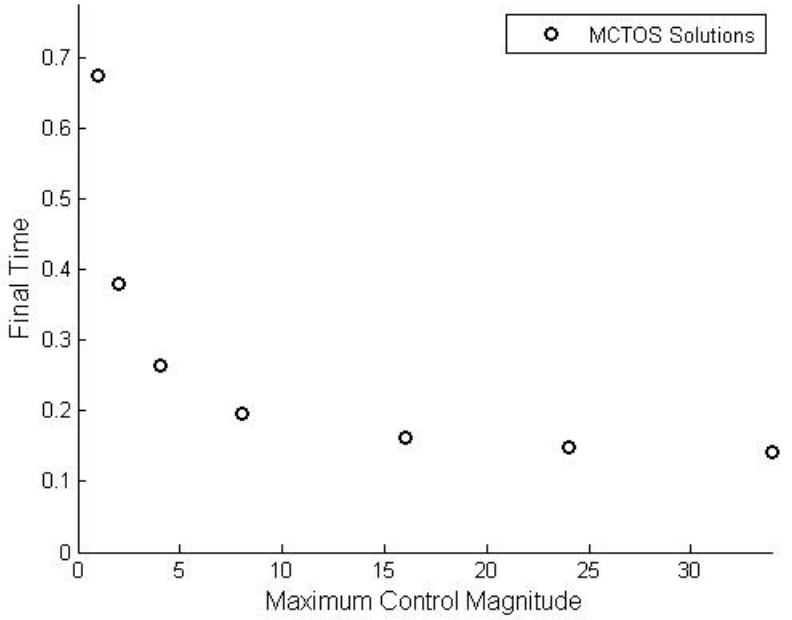


Figure 6.2: Final time vs. maximum control amplitude. The circles give the minimum time required to compute the one qubit Hadamard gate with a given upper bound on the control input.

6.3 Visual Interpretation of the Optimization Surface

To give a visual interpretation of the surface Φ, Figure 6.3 shows a plot of this surface in the case where the first interval ξ_1 has both controls set to the maximum amplitude and where the interval ξ_2 has only the first control set to the maximum amplitude, while the second control has magnitude zero. By inspection, we see that the level set $\Phi(\xi) = 0$ is given by only three disconnected points in the displayed region. The closest one to the origin corresponds to the answer from Table 6.2. There is another at $(\xi_1, \xi_2) = (0, 0.5303)$, corresponding to the answer from Table 6.1. The last point of the level set in this region is further away from the origin than either of the previous two and is equivalent to a periodic shift of the first one.

Note that the surface $\Phi(\xi)$ is in general different than the one in Figure 6.3 and depends on the number of intervals and the order of control switches. In particular, this plot does not prove that the solution from Table 6.2 is the minimum time solution, just that it is the minimum time solution for this number and order of switches.

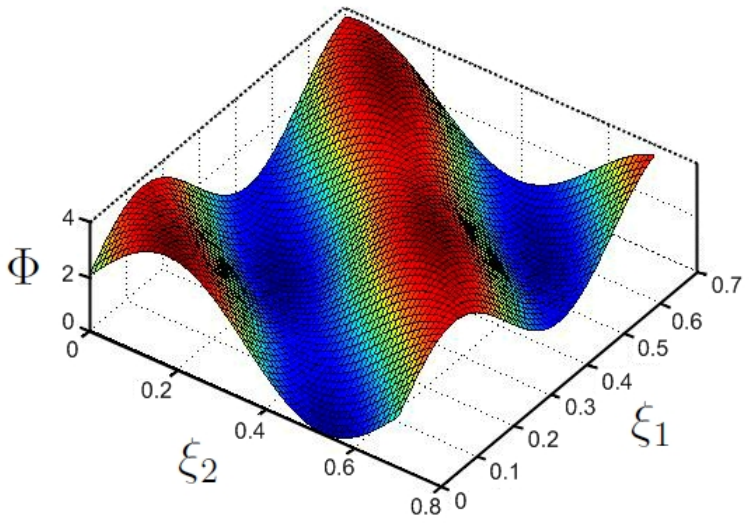


Figure 6.3: Two-dimensional optimization surface. The surface gives the value of Φ over the time intervals ξ_1 and ξ_2. The interval ξ_1 gives the length of time for which both controls take on their maximum values, while in the interval ξ_2 only v_1 takes on its maximum value.

Chapter 7

Discussion

7.1 Summary

In this thesis we presented the problem of time-optimal control of the Schrödinger operator equation

$$\dot{X} = -iH(v)X. \tag{7.1}$$

We established the quantum mechanical background behind the dynamics and presented a summary of related results from the literature. The well-known Lie algebra rank condition for the controllability from [29] was presented in Chapter 3 along with an algorithm which can be used to determine the controllability of a particular system.

Chapter 4 discussed the application of the Pontryagin Maximum Principle to the problem of time-optimal unitary synthesis. In particular, this principle was generalized to be applicable to complex matrix systems and then specialized to quantum mechanical systems such as (7.1). The example of a one qubit system was considered to solidify the theory in Section 4.4. It was demonstrated that the control function associated to a singular interval must be identically zero on that interval. This implies that the control law is bang-bang, since the lower bound of the control input we are using is zero.

In chapers 5 and 6 we presented and implemented a numerical optimization method to compute bang-bang controls for (7.1). The algorithm was applied to find the minimum time required to produce a one qubit Hadamard gate with bounded controls for several possible values of the maximum control amplitude. We also investigated whether or not the minimum time seemed to decrease to zero as the control amplitude was increased.

7.2 Open Questions and Future Work

The results mentioned in the previous section are the preliminary investigations into the time-optimal contol of (7.1). There are still many avenues for future research

on this problem which were not covered by this thesis.

On the theoretical front, many questions regarding the application of the Maximum Principle to the Schrödinger equation persist. In particular, the types of extremals that are possible in the general case remains an open question as discussed in Section 4.7. Even in the one qubit case we have not shown that singular intervals do or do not exist. We have only shown that if they do, the associated control must be zero on that interval. A similar result for systems with arbitrary qubits would go a long way to putting the search for bang-bang controls on a solid mathematical footing.

Additionally, the question of the existence of abnormal extremals has not been carefully considered in this thesis. In [4], this question is examined in a few special cases for control systems on real Lie groups. It has not been determined whether or not these results may be applied to system (7.1).

A hugely important but likely very difficult problem is to determine the constant, traceless, skew-Hermitian matrix M which arises from the Maximum Principle and the right-invariance of system (7.1) in Section 4.4. Solving this problem would essentially amount to a complete solution of the time-optimal control problem since it would give analytic expressions for the switching functions in terms of the exponential matrix of the quantum Hamiltonian. Determining the switching times would then become the more simple matter of finding the roots of the switching functions.

In the absence of such a complete solution, learning more information about the switching functions and the switching times would be extremely useful. In the single qubit case, we could find the switching functions explicitly and were therefore able to conclude that an infinite number of switches was not possible. We could also set an upper bound on the length of time between switches for a given control. Since it is unlikely that it will be possible to determine closed form solutions for the switching functions in problems with an arbitrary number of qubits, new methods for finding information about the switching times are required.

In the area of computation of time optimal controls there is much room for improvement as well. While we briefly discussed the properties of the terminal surface in Chapter 5, a better understanding is still necessary. In particular, investigating the periodic nature of the solutions in systems with more than one qubit would be helpful for reducing the total time required for generating the desired operator.

It is also unclear whether or not the terminal surface is even path-connected. That is, given any two points ξ^1 and ξ^2 which satisfy $\Phi(\xi) = 0$ is it possible to find a continuous function $f(x) : [0, 1] \mapsto \mathbb{R}^k$ such that $f(0) = \xi^1$ and $f(1) = \xi^2$ and $\Phi(f(x)) = 0$ for all x in $[0, 1]$. If the terminal surface is not path-connected then it makes optimization challenging because moving around on the terminal surface may not produce the optimal solution. Being able to reduce the period of intervals could be a way to circumvent this problem if a good period-reducing method could be found. This might allow one to jump between different segments of the terminal surface, hopefully arriving at a segment which contains the optimal

solution. Figure 7.1 shows what reducing the period might look like for a periodic, disconnected terminal surface in a two-dimensional ξ-space.

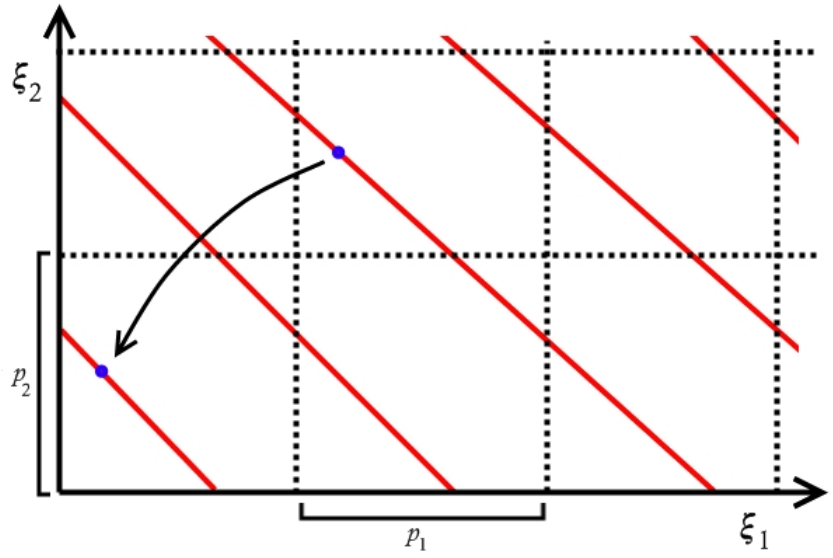


Figure 7.1: Period reduction. The dashed lines represent integer multiples of the periods. The diagonal line is the terminal surface (the points which satify $\Phi(\xi) = 0$). In this example this is just multiple parallel lines. The arrow shows how an ideal period-reducing algorithm could find an equivalent point on the terminal surface by writing the ξ vector modulo the period.

Another aspect of the problem to consider is the implementation of a more suitable algorithm for moving along the terminal surface. The current code has been able to find local minima for one qubit, but has not yet been successful for multiple qubit systems. This is likely because the terminal surface becomes much more complicated for systems with more qubits. The current method of finding a linearization and then solving the linear programming problem may not be a good choice in regions where the terminal surface is highly nonlinear. In those cases, a solution of the linear problem may not be useful as it could be very far from the solution to the nonlinear problem.

One option to circumvent this problem would be to find a quadratic estimate of the terminal surface instead of a linear one. A quadratic approximation would preserve more of the characteristics of the terminal surface and the solution to the quadratic optimization problem should be closer. A drawback of this would be the extra computation required to calculate the approximation.

Alternatively, it might be possible to do better using only the linearization. Keeping in mind that the goal is to minimize the final time, we could look for the direction along the terminal surface in which the final time decreases the fastest. If we didn't care about staying on the terminal surface, the best direction to move in to decrease the final time is given by $-\xi$ no matter where we are in the optimization space. This is just the vector which points from ξ towards the origin. The projection

of this vector onto the terminal surface therefore gives the best direction to move along the terminal surface to decrease the final time. A local minimum or maximum occurs when the projection of $-\xi$ onto the terminal surface is the zero vector. When the projection is not the zero vector, a decrease in the terminal time could be guaranteed for a sufficiently small step size. When the projection is small, this indicates that we might be close to a local minimum and that solving the linear programming problem would be beneficial. Either of these two approaches could potentially improve the results of the code for multiple qubit systems.

Another very important consideration is the overall efficiency of the algorithm. Since the size of matrices in the problem grow exponentially with the number of qubits there is no room for wasted effort. While optimizing the code is not necessarily as exciting as optimizing the control sequence, it is necessary for the algorithm to be useful for working with systems of multiple qubits.

Once the optimization algorithm is sufficiently robust for multiple qubit systems, it would be interesting to compare the results with other similar programs such as the GRAPE algorithm. This algorithm would be a viable alternative to GRAPE if it were either more efficient or if it could find shorter control sequences to produce unitary operators.

Appendix A

Useful Properties and Results

Let the capital roman letters A and B denote elements of $\mathbb{C}^{N\times N}$ and let α be a complex scalar.

Theorem A.1. *The dagger operation defined by*

$$A^\dagger = \bar{A}^T$$

satisfies the following properties:

1. $$(AB)^\dagger = B^\dagger A^\dagger$$

2. $$(\alpha A)^\dagger = \bar{\alpha} A^\dagger$$

3. $$(A+B)^\dagger = A^\dagger + B^\dagger$$

4. $$\left(\frac{dA}{dt}\right)^\dagger = \left(\frac{dA^\dagger}{dt}\right)$$

Proof. The proof of these results follow directly from the definitions of the conjugate and transpose operations. □

Theorem A.2. *The scalar-valued trace operation is defined by*

$$\mathrm{Tr}(A) = \sum_{j=1}^{N} (a_{jj}),$$

where a_{jk} is the element in the j^{th} row and k^{th} column of A. The trace operation satisfies the following properties:

1.

$$\mathrm{Tr}\left(\sum_{k=1}^{m}\alpha_k A_k\right) = \sum_{k=1}^{m}\alpha_k \mathrm{Tr}\left(A_k\right),$$

2.

$$\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$$

Proof. The first result follows from induction and the properties

$$\mathrm{Tr}(A+B) = \mathrm{Tr}(A) + \mathrm{Tr}(B)$$

and

$$\mathrm{Tr}(\alpha A) = \alpha \mathrm{Tr}(A).$$

The second property is the well-known cyclic property of trace. □

Theorem A.3. *The matrix inner product is defined by*

$$\langle A, B\rangle = \mathrm{Tr}(A^\dagger B).$$

Let C and X be real, $N \times N$ matrices. The derivative with respect to X of the inner product of C and X is

$$\frac{d}{dX}\langle C, X\rangle = \frac{d}{dX}\langle X, C\rangle = C.$$

Proof. The proof follows from the definition of the trace operation and equation (4.7). □

Theorem 4.1. *Let P and Q be complex $N \times N$ matrices such that*

$$\begin{aligned} P &= P_R + iP_I \\ Q &= Q_R + iQ_I. \end{aligned}$$

Let the corresponding real $2N^2$-dimensional vectors p and q be defined by

$$p = \mathrm{vec}\left(\begin{bmatrix} P_R & P_I \end{bmatrix}\right)$$

and

$$q = \mathrm{vec}\left(\begin{bmatrix} Q_R & Q_I \end{bmatrix}\right)$$

The following statement holds

$$\mathrm{Re}\,\langle P, Q\rangle_{\mathbb{C}^{N\times N}} = \langle p, q\rangle_{\mathbb{R}^{2N^2}}.$$

Proof. Define the j^{th} column of P_R by p_{R_j}, and likewise for P_I, Q_R and Q_I. In terms of the vectors p_{R_j}, p_{I_j}, q_{R_j} and q_{I_j},

$$p = \operatorname{vec}\left(\begin{bmatrix} P_R & P_I \end{bmatrix}\right) = \begin{bmatrix} p_{R_1} \\ \vdots \\ p_{R_N} \\ p_{I_1} \\ \vdots \\ p_{I_N} \end{bmatrix}, \quad q = \operatorname{vec}\left(\begin{bmatrix} Q_R & Q_I \end{bmatrix}\right) = \begin{bmatrix} q_{R_1} \\ \vdots \\ q_{R_N} \\ q_{I_1} \\ \vdots \\ q_{I_N} \end{bmatrix},$$

so their inner product is given by

$$\begin{aligned} \langle p, q \rangle_{\mathbb{R}^{2N^2}} &= \begin{bmatrix} p_{R_1}^T & \cdots & p_{R_N}^T & p_{I_1}^T & \cdots & p_{I_N}^T \end{bmatrix} \begin{bmatrix} q_{R_1} \\ \vdots \\ q_{R_N} \\ q_{I_1} \\ \vdots \\ q_{I_N} \end{bmatrix} \\ &= \sum_{j=1}^{N} p_{R_j}^T q_{R_j} + \sum_{j=1}^{N} p_{I_j}^T q_{I_j} \end{aligned}$$

The real part of the inner product between P and Q is given by

$$\begin{aligned} \operatorname{Re}\langle P, Q \rangle_{\mathbb{C}^{N\times N}} &= \operatorname{Re}\left\{\operatorname{Tr}\left(P^\dagger Q\right)\right\} \\ &= \operatorname{Re}\left\{\operatorname{Tr}\left((P_R + iP_I)^\dagger (Q_R + iQ_I)\right)\right\} \\ &= \operatorname{Re}\left\{\operatorname{Tr}\left(P_R^T Q_R + P_I^T Q_I + i(P_R^T Q_I - P_I^T Q_R)\right)\right\} \\ &= \operatorname{Tr}\left(P_R^T Q_R + P_I^T Q_I\right) \\ &= \operatorname{Tr}\left(\begin{bmatrix} p_{R_1}^T \\ \vdots \\ p_{R_N}^T \end{bmatrix} \begin{bmatrix} q_{R_1} & \cdots & q_{R_N} \end{bmatrix} + \begin{bmatrix} p_{I_1}^T \\ \vdots \\ p_{I_N}^T \end{bmatrix} \begin{bmatrix} q_{I_1} & \cdots & q_{I_N} \end{bmatrix}\right) \\ &= \sum_{j=1}^{N} p_{R_j}^T q_{R_j} + \sum_{j=1}^{N} p_{I_j}^T q_{I_j} \\ &= \langle p, q \rangle_{\mathbb{R}^{2N^2}} \end{aligned}$$

which completes the proof. □

Appendix B

Groups and Differential Geometry

This appendix contains a summary of the key definitions needed for defining Lie groups and Lie algebras as well as a proof that the unitary matrices are a Lie group.

Definition B.1. Group ([23, §2]) *A group $(\mathcal{G}, *)$ is a set $\mathcal{G}$ together with the binary operation $*$ which satisfies the following axioms:*

1. *Closure: For all g and h in $\mathcal{G}$, the product $g * h$ is in $\mathcal{G}$*
2. *Associativity: For all g, h and f in $\mathcal{G}$,*
$$g * (h * f) = (g * h) * f$$
3. *Identity element: There exists an element e in $\mathcal{G}$ such that for all g in $\mathcal{G}$,*
$$eg = ge = g$$
4. *Inverse element: For each g in $\mathcal{G}$ there exists g^{-1} in $\mathcal{G}$ such that*
$$gg^{-1} = g^{-1}g = e$$

Remark B.1. The binary operation $*$ is called the *group operation*.

A differentiable manifold is a space which, locally, resembles Euclidean space ($\mathbb{R}^n$) and which has a suitable structure for applying the tools of calculus. The following definitions formalize this concept.

Definition B.2. *A topological space (Z, T) is a set Z, with a collection of subsets T of Z which satisfy the following axioms:*

1. *T contains both Z and $\{\emptyset\}$*
2. *The union of collection of sets in T is also in T*

3. The finite intersection of any collection of sets in T is also in T

The elements of Z are usually called points. The sets in T are called open. The complement of an open set is closed. Sets may be open, closed, both or neither.

Definition B.3. ([28, §1]) *A Hausdorff space is a topological space which possesses the property that any two points in the space x and y have neighbourhoods U and V, respectively, such that $U \cap V = \{\emptyset\}$.*

Definition B.4. ([11]) *A n-dimensional topological manifold M^n is a Hausdorff space with a countable basis for the topoplogy which is locally homeomorphic to $\mathbb{R}^n$, that is, for each point $p \in M^n$ there exists an open neighbourhood $U \subset M^n$ of p and a homeomorphism*

$$h : U \to U'$$

onto an open set $U' \subset \mathbb{R}^n$.

Definition B.5. ([11]) *The homeomorphisms $\hbar$ from the previous definition are called charts of M^n. A collection of charts $\{\hbar_\alpha | \alpha \in A\}$ with domains U_α is called an atlas for M^n if the union of the domains is all of M^n. On subsets where chart domains overlap, for example say on $U_j \cap U_k$, we can define a chart transformation between charts as $\hbar_k \circ \hbar_j^{-1}$.*

Definition B.6. ([11]) *An atlas of a manifold is called* differentiable *if all of its chart transformations are differentiable. An atlas is called* maximal *if is not contained in any other atlas. A maximal differentable atlas is called a differentiable structure on the topological manifold.*

Definition B.7. ([64, 11, 26, 28]) *A n-dimensional differentiable manifold is a topological manifold with a differentiable structure and is denoted $(M, \mathcal{D})$ or simply M for short.*

Lie groups are defined in Definition 3.8. Note that Lie groups are n-dimensional differentiable manifolds. This means that Lie groups are intrinsically finite-dimensional. The following is a sketch of the proof of Theorem 3.4 .

Theorem 3.4. *The set of $N \times N$ unitary matrices, $\mathcal{U}(N)$, are a Lie group.*

Proof. We first show that the group definition is satisfied.

1. Inverse: For any unitary X, its inverse is given by $X^\dagger$, by definition of a unitary matrix. Furthermore,

   $$(X^\dagger)^\dagger X^\dagger = X X^\dagger = \mathbb{1}_N,$$

   so $X^\dagger$ is unitary as well. Therefore every unitary matrix has a unitary inverse.

2. Identity: Clearly $\mathbb{1}_N \mathbb{1}_N^\dagger = \mathbb{1}_N \mathbb{1}_N = \mathbb{1}_N$, and so the standard identity matrix of dimension $N \times N$ is a unitary matrix.

3. Associativity: Associativity follows directly from the properties of matrices.

4. Closure: The unitary matrices must also be closed under matrix multiplication, and indeed we have that

$$\begin{aligned}(XY)^\dagger(XY) &= (Y^\dagger X^\dagger)(XY)\\ &= Y^\dagger(X^\dagger X)Y\\ &= Y^\dagger Y\\ &= \mathbb{1}_N,\end{aligned}$$

   if X and Y are unitary. Thus unitarity is closed under matrix multiplication.

By definition, the set of $N \times N$ unitary matrices is a group. We also need to show that the group operations (matrix multiplication and inversion) are analytic. This is beyond the scope of this report and so we state it as a fact without demonstration.

To complete the proof we need to show that the unitary matrices form a differential manifold. This proof is only sketched out since the full version is, again, beyond the scope of this report.

We use the fact that the unitary matrices of dimension $N \times N$ are a subset of the complex $N \times N$ general linear matrices, $\mathcal{GL}(N, \mathbb{C})$. $\mathcal{GL}(N, \mathbb{C})$ has elements in $\mathbb{C}^{N\times N}$, a space which is homeomorphic to $\mathbb{R}^{2N^2}$. Therefore it is easily shown to be a differentiable manifold.

$\mathcal{U}(N)$ is a closed subset of $\mathcal{GL}(N, \mathbb{C})$ and therefore a submanifold by definition. This proves that the unitary matrices are a differential manifold. In combination with the fact that they are a group, this proves that the unitary matrices are a Lie group. □

Theorem B.1. *The Pontryagin Hamiltonian*

$$\mathscr{H} = \lambda_0 + \mathrm{Re}\left\langle M, -iX^\dagger(t)H_jX(t)\right\rangle,$$

where M is an arbitrary complex matrix is equal to

$$\mathscr{H} = \lambda_0 + \mathrm{Re}\left\langle A, -iX^\dagger(t)H_jX(t)\right\rangle$$

for some A in $\mathfrak{su}(N)$.

Proof. We recall that any matrix can be written uniquely as the sum of a Hermitian matrix and a skew-Hermitian matrix.

Let $M = S + S^\perp$, where $S^\perp$ is a Hermitian matrix, and S is a skew-Hermitian matrix.

$$\begin{aligned}&\mathrm{Re}\left\{\left\langle M, -iX^\dagger(t)H_jX(t)\right\rangle\right\}\\ =\ &\mathrm{Re}\left\{\left\langle S^\perp + S, -iX^\dagger(t)H_jX(t)\right\rangle\right\}\\ =\ &\mathrm{Re}\left\{\left\langle S^\perp, -iX^\dagger(t)H_jX(t)\right\rangle\right\} + \mathrm{Re}\left\{\left\langle S, -iX^\dagger(t)H_jX(t)\right\rangle\right\}\end{aligned}$$

We now consider the first term, with the Hermitian matrix $S^\perp$.

$$\begin{aligned}
\mathrm{Re}\left\{\left\langle S^\perp, -iX^\dagger(t)H_jX(t)\right\rangle\right\} &= \mathrm{Re}\left\{\mathrm{Tr}\left((S^\perp)^\dagger(-i)X^\dagger(t)H_jX(t)\right)\right\} \\
&= \mathrm{Re}\left\{\mathrm{Tr}\left(S^\perp(-i)X^\dagger(t)H_jX(t)\right)\right\} \\
&= \mathrm{Re}\left\{\mathrm{Tr}\left(-iX^\dagger(t)H_jX(t)S^\perp\right)\right\} \\
&= \mathrm{Re}\left\{\left\langle iX^\dagger(t)H_jX(t), S^\perp\right\rangle\right\} \\
&= \mathrm{Re}\left\{\overline{\left\langle S^\perp, iX^\dagger(t)H_jX(t)\right\rangle}\right\} \\
&= -\mathrm{Re}\left\{\overline{\left\langle S^\perp, -iX^\dagger(t)H_jX(t)\right\rangle}\right\}
\end{aligned}$$

The final equality gives us that

$$\begin{aligned}
0 &= \mathrm{Re}\left\{\left\langle S^\perp, X^\dagger(t)(-iH_j)X(t)\right\rangle + \overline{\left\langle S^\perp, X^\dagger(t)(-iH_j)X(t)\right\rangle}\right\} \\
\Rightarrow 0 &= \mathrm{Re}\left\{2\mathrm{Re}\left\{\left\langle S^\perp, X^\dagger(t)(-iH_j)X(t)\right\rangle\right\}\right\} \\
\Rightarrow 0 &= \mathrm{Re}\left\{\left\langle S^\perp, X^\dagger(t)(-iH_j)X(t)\right\rangle\right\}
\end{aligned}$$

Since the Hermitian part of the matrix M does not contribute to the inner product we have

$$\mathrm{Re}\left\{\left\langle M, -iX^\dagger(t)H_jX(t)\right\rangle\right\} = \mathrm{Re}\left\{\left\langle S, -iX^\dagger(t)H_jX(t)\right\rangle\right\},$$

for some skew-Hermitian S.

The next step is to show that only the traceless part of S contributes to the inner product. Denote the trace of S by $\mathrm{Tr}(S) = is$, where s is a real, non-negative number (it is easy to verify that the trace of a skew-Hermitian matrix is purely imaginary). This means that the matrix $A = S - \frac{is}{N}\mathbb{1}$ has trace 0. Now consider the inner product

$$\left\langle S - \frac{is}{N}\mathbb{1}, X^\dagger(t)(-iH_j)X(t)\right\rangle = \left\langle S, X^\dagger(t)(-iH_j)X(t)\right\rangle - \frac{is}{N}\left\langle \mathbb{1}, X^\dagger(t)(-iH_j)X(t)\right\rangle.$$

We now show that the second term vanishes.

$$\begin{aligned}
-\frac{is}{N}\left\langle \mathbb{1}, X^\dagger(t)(-iH_j)X(t)\right\rangle &= -\frac{is}{N}\mathrm{Tr}\left(\mathbb{1}^\dagger X^\dagger(t)(-iH_j)X(t)\right) \\
&= -\frac{is}{N}\mathrm{Tr}\left(X^\dagger(t)(-iH_j)X(t)\right) \\
&= -\frac{is}{N}\mathrm{Tr}\left(X(t)X^\dagger(t)(-iH_j)\right) \\
&= -\frac{is}{N}\mathrm{Tr}\left((-iH_j)\right),
\end{aligned}$$

where we have used the cyclic property of trace, as well as the fact that $X(t)$ is unitary.

Finally, we note that the $-iH_j$ matrices are all elements of $\mathfrak{su}(N)$, and therefore have zero trace. From above this gives us that

$$\left\langle S - \frac{is}{N}\mathbb{1}, -iX^\dagger(t)H_jX(t)\right\rangle = \left\langle S, -iX^\dagger(t)H_jX(t)\right\rangle.$$

So the traceless matrix A yields the same inner product as the matrix S. This means the inner product only depends on the traceless part of the matrix. We can therefore write the Pontryagin Hamiltonian as

$$\mathscr{H} = \lambda_0 + \operatorname{Re}\left\langle A, -iX^\dagger(t)H_jX(t)\right\rangle.$$

where A is a traceless skew-Hermitian matrix, which completes the proof. □

The previous result showed that we could replace the arbitrary matrix M with an arbitrary special unitary matrix A. The next result shows that M must actually be an element of $\mathfrak{su}(N)$ in the first place.

Theorem 4.5. *The matrix M from the Pontryagin Hamiltonian*

$$\mathscr{H} = \lambda_0 + \operatorname{Re}\left\langle M, -iX^\dagger(t)H_jX(t)\right\rangle$$

is an element of $\mathfrak{su}(N)$.

The full proof of this result is beyond the scope of this thesis, but we give an idea as to why it holds. Similar results appears in [4, §3] and in [68].

First, we discuss some results from geometric control theory. In particular, [28] generalizes the Maximum Principle to systems which evolve on manifolds of which systems evolving on Lie groups are a special case. From the geometric viewpoint, the costate is not just the solution to the equation

At each point X on a k-dimensional manifold $\mathcal{M}$ we can define the tangent space $T_X\mathcal{M}$ which is the space of all tangent vectors to the point X. The tangent space is also a k-dimensional manifold. The set of tangent spaces for all points X in $\mathcal{M}$ forms a vector space as well and is called the tangent bundle $T\mathcal{M}$. The tangent bundle is composed of all pairs of the form $(X, Y) \in \mathcal{M} \times T_X\mathcal{M}$. The dual space of the tangent bundle is a called the cotangent bundle and is denoted $T^*\mathcal{M}$.

In the case where $\mathcal{M} = \mathcal{G}$, a Lie group, the tangent space is given by the Lie algebra of $\mathcal{G}$ denoted $\mathcal{L}$. The tangent bundle is given by pairs $(X, Y) \in \mathcal{G} \times \mathcal{L}$. The cotangent bundle for systems on Lie groups can be realized as $T^*\mathcal{G} = \mathcal{G} \times \mathcal{L}^*$, where $\mathcal{L}^*$ is the dual space of the Lie algebra $\mathcal{L}$ [28, §12.1.1].

For Lie groups, we can define the adjoint action on the Lie algebra by

$$\operatorname{Ad}_g(L) = gLg^{-1},$$

for all $g \in \mathcal{G}$ and $L \in \mathcal{L}$. The dual of the adjoint action is called the co-adjoint action and is denoted $\operatorname{Ad}^*_g(K)$. The action defines the co-adjoint orbits through the point $K \in \mathcal{L}^*$. The costate Λ can be expressed in terms of the adjoint orbit which in turn implies M must be an element of $\mathcal{L}$. For more information see [28, §12].

Appendix C

Costate Algebra

Theorem 4.3. *The costate dynamics evolve under the same equation as the state dynamics (4.10), that is*

$$\frac{d\Lambda(t)}{dt} = -iH(v)\Lambda(t),$$

though the initial condition Λ_0 may be different from X_0.

Proof. By the CMMP, the costate dynamics are given by

$$\frac{d\Lambda(t)}{dt} = -\frac{d\mathscr{H}}{dX} = -\left(\frac{\partial\mathscr{H}}{\partial X_R} + i\frac{\partial\mathscr{H}}{\partial X_I}\right) \tag{C.1}$$

Recall that the Hamiltonian $\mathscr{H}$ is

$$\mathscr{H} = \lambda_0 + \operatorname{Re}\langle\Lambda(t), -iH(t)X(t)\rangle .$$

The expression $\operatorname{Re}\langle\Lambda, -iHX\rangle$ can be simplified by breaking all the matrices into their real and imaginary parts as

$$\begin{aligned}\operatorname{Re}\langle\Lambda, -iHX\rangle &= \operatorname{Re}\langle\Lambda_R + i\Lambda_I, -i(H_R + iH_I)(X_R + iX_I)\rangle \\ &= \operatorname{Re}\langle\Lambda_R + i\Lambda_I, H_RX_I - iH_RX_R + H_IX_R + iH_IX_I\rangle\end{aligned}$$

By the definition of the inner product

$$\operatorname{Re}\langle\Lambda, -iHX\rangle = \operatorname{Re}\left\{\operatorname{Tr}\left((\Lambda_R^T - i\Lambda_I^T)(H_RX_I - iH_RX_R + H_IX_R + iH_IX_I)\right)\right\}$$

Expanding and taking the real part of the previous expression we obtain

$$\operatorname{Re}\langle\Lambda, -iHX\rangle = \operatorname{Tr}\left((\Lambda_R^TH_I - \Lambda_I^TH_R)X_R\right) + \operatorname{Tr}\left((\Lambda_R^TH_R + \Lambda_I^TH_I)X_I\right).$$

Using the definition of inner product again the expression becomes

$$\operatorname{Re}\langle\Lambda, -iHX\rangle = \left\langle(\Lambda_R^TH_I - \Lambda_I^TH_R)^T, X_R\right\rangle + \left\langle(\Lambda_R^TH_R + \Lambda_I^TH_I)^T, X_I\right\rangle$$

This simplifies to

$$\mathrm{Re}\,\langle \Lambda, -iHX\rangle = \left\langle H_I^T\Lambda_R - H_R^T\Lambda_I, X_R\right\rangle + \left\langle H_R^T\Lambda_R + H_I^T\Lambda_I, X_I\right\rangle.$$

Recall that $H = H_R + iH_I$ is a Hermitian matrix. This implies that

$$H^\dagger = H_R^T - iH_I^T = H_R + iH_I = H,$$

which in turn implies that $H_R = H_R^T$ and $H_I = -H_I^T$. Using this we obtain

$$\mathrm{Re}\,\langle \Lambda, -iHX\rangle = \langle -H_I\Lambda_R - H_R\Lambda_I, X_R\rangle + \langle H_R\Lambda_R - H_I\Lambda_I, X_I\rangle.$$

Substituting into (C.1) we get

$$\frac{d\Lambda}{dt} = -\frac{\partial}{\partial X_R}\langle -H_I\Lambda_R - H_R\Lambda_I, X_R\rangle - i\frac{\partial}{\partial X_I}\langle H_R\Lambda_R - H_I\Lambda_I, X_I\rangle.$$

Applying Theorem A.3 gives

$$\frac{d\Lambda}{dt} = (H_I\Lambda_R + H_R\Lambda_I) + i(-H_R\Lambda_R + H_I\Lambda_I).$$

We now factor the expression to obtain

$$\frac{d\Lambda}{dt} = -i(H_R + iH_I)(X_R + iH_I),$$

which is the same as

$$\frac{d\Lambda(t)}{dt} = -iH(v)\Lambda(t).$$

This dynamics equation is the same as the dynamics equation for $X(t)$ given by (4.10). □

Appendix D

Controllability Algorithm

```
function [r] = cont(Ain)
% Ain is a vector of basis nxn matrices
% r is the dimension of the Lie algebra

A = Ain;   N = size(A,1);
c = size(A,3);  last = 0;
B=zeros(2*N^2,0);    r=0;

while c>0 && r<(N^2-1);
   c=0;  oldlast = last;  last = size(A,3);

   for jj=1:last-1
        s = max(jj,oldlast);
        for kk=s+1:last
            Atmp = com(A(:,:,jj),A(:,:,kk));
%Calulate the previous rank of the B matrix then add the
%new element and Calculate the new rank of the B matrix
            rprev = rank(B);
            AR = real(Atmp); AI = imag(Atmp);
            B(:,end+1) = [reshape(AR,N^2,1); reshape(AI,N^2,1)];
            r = rank(B);
%If matrix does not increase span, remove it
            if r == rprev
                B = B(:,1:end-1);
            else
                c = c + 1;  A(:,:,end+1) = Atmp;
            end
        end
    end
end
```

Appendix E

Converting Between Numeric Representations of Intervals

In Section 5.4 it is stated that there exists an invertible transformation between the interval matrix α and the vector of consecutive interval lengths ξ. In this appendix we explain how to construct such a transformation. We first demonstrate with the example given in Figure E.1 before generalizing.

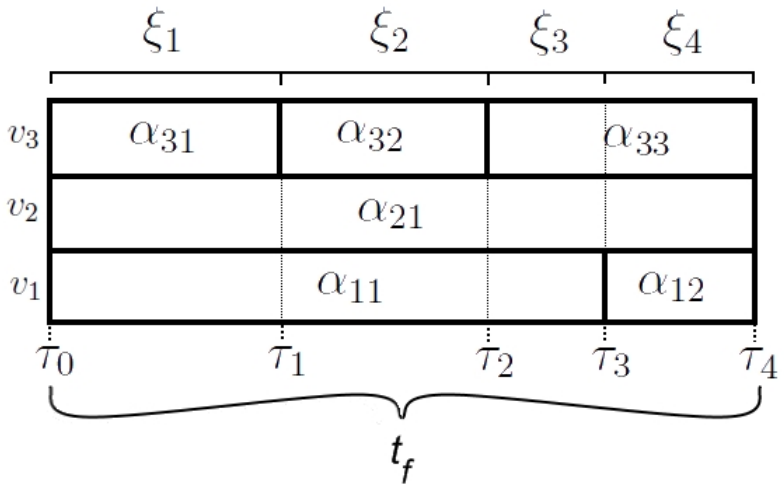


Figure E.1: Intervals for multiple controls. The horizontal "axis" denotes time while the controls are stacked on the vertical axis.

The example in Figure E.1 has three controls and five switching times. The first switching time τ_0 is the initial time and the final switching time τ_4 corresponds to the final time t_f. The vector of interval lengths ξ is

$$\xi = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{bmatrix}.$$

The vector $\mathcal{C}$ which keeps track of which control switches at each switching time is

$$\mathcal{C} = \begin{bmatrix} 3 & 3 & 1 & 1 \end{bmatrix}.$$

Note that we have chosen the last switch to correspond to the first control. We could have alternatively chosen control two or three.

The vectors α_j which correspond to the interval lengths between switches of the same control are

$$\alpha_1 = \begin{bmatrix} \alpha_{11} \\ \alpha_{12} \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} \alpha_{21} \end{bmatrix}, \quad \alpha_3 = \begin{bmatrix} \alpha_{31} \\ \alpha_{32} \\ \alpha_{33} \end{bmatrix},$$

and may be written as a single matrix α by

$$\alpha = \begin{bmatrix} \alpha_{11} & \alpha_{21} & \alpha_{31} \\ \alpha_{12} & 0 & \alpha_{32} \\ 0 & 0 & \alpha_{33} \end{bmatrix}.$$

The vector κ which stores the number of intervals for each control is

$$\kappa = \begin{bmatrix} 2 & 1 & 3 \end{bmatrix}$$

for this example.

Note that there are six intervals from the matrix viewpoint but only four from the vector viewpoint. This discrepancy would make it impossible to define an invertible transformation between the two. To resolve this issue, we add two zero length intervals to ξ corresponding to controls two and three. This gives

$$\xi = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \end{bmatrix} \quad \text{and} \quad \mathcal{C} = \begin{bmatrix} 3 & 3 & 1 & 1 & 2 & 3 \end{bmatrix},$$

where ξ_5 and ξ_6 are both zero. We can think of the last two intervals as being the formal expression of the fact that all three controls stop at the same time.

Alternatively, we could have rewritten the final intervals associated to controls two and three (α_{21} and α_{33}, respectively) in terms of the other elements of α since they are constrained by the fact that all of the controls must end at the same time. To avoid inequality constraints in the LP problem, we choose to use the former idea of adding zero length intervals to ξ.

We now show how to construct α from ξ and $\mathcal{C}$. We start by computing α_1. The first time the first control switches is at the third switching time. The length

of α_{11} is therefore the sum of all the intervals of ξ up to and including the third. This gives

$$\alpha_{11} = \xi_1 + \xi_2 + \xi_3.$$

The second and final switch associated to the first control occurs at the very next switching time. This implies that

$$\alpha_{12} = \xi_4.$$

In a similar fashion, we compute α_{21} by summing the intervals up to and including the fifth interval (because the first time control two switches is at the fifth switching time). This gives

$$\alpha_{21} = \xi_1 + \xi_2 + \xi_3 + \xi_4 + \xi_5.$$

Finally, we calculate the three intervals associated to the third control and get

$$\begin{aligned} \alpha_{31} &= \xi_1, \\ \alpha_{32} &= \xi_2, \\ \alpha_{33} &= \xi_3 + \xi_4 + \xi_5 + \xi_6. \end{aligned}$$

We can express this in matrix form as

$$\begin{bmatrix} \alpha_{11} \\ \alpha_{12} \\ \alpha_{21} \\ \alpha_{31} \\ \alpha_{32} \\ \alpha_{33} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \end{bmatrix}$$

The vector on the left of the equality contains the stacked columns of α. In Section 5.4 we denoted this vector by σ. The matrix of ones and zeros is the matrix representation of the linear transformation between ξ and σ. In the terminology of Section 5.4 this is the transformation B^{-1}.

In general, the matrix B^{-1} can be determined using the Matlab function `getBInv`, which is implemented as follows.

```
function [Binv] = getBInv(switchOrder,kappa,m)

%Initialize variables
Binv = zeros([sum(kappa),sum(kappa)]);
currentRow = 1;

%For each control:
for control = 1:m
    prevIndex = 1;
```

```
%For each row associated to the control
    for rr = 1:kappa(control)

%Find the next switch associated to this control
        nextIndex = prevIndex;
        while switchOrder(nextIndex)~= control
            nextIndex = nextIndex + 1;
        end

%Set the appropriate entries to 1
        Binv(currentRow, prevIndex:nextIndex) = 1;
        prevIndex = nextIndex + 1;
        currentRow = currentRow + 1;
    end
end
```

Let $\tilde{\kappa} = \sum_{j=1}^{m} \kappa_j$. The matrix B^{-1}, denoted `Binv` in the code, is a $\tilde{\kappa} \times \tilde{\kappa}$ real matrix for which each entry is either a zero or a one. The j^{th} control has κ_j rows associated to it. To compute the rows associated to a given control, start at the first column of the first associated row. Set the entry of the current row and column to one. Check if the entry of the corresponding column of $\mathcal{C}$ (the switching order vector) is number of the current control. If so then move to the next row. If the next row corresponds to a different control then restart at the first column, otherwise more to the next column. Repeat from where the current entry is set to one until all $\tilde{\kappa}$ rows have been filled in. All entries that have not been set to one should be zeros.

Theorem E.1. *The matrix B^{-1} constructed from the previous algorithm is invertible.*

Proof. The switching order vector $\mathcal{C}$ has the form

$$\mathcal{C} = \left[\; c_1, c_2, \ldots, c_{\tilde{\kappa}} \; \right].$$

For each entry c_j in $\mathcal{C}$, there is a corresponding row of B^{-1} for which the j^{th} column is one and all the entries to the right are zeros. This is because in some row corresponding to the c_j^{th} control the preceding algorithm sets j^{th} column of that row to one and then moves to the next column, leaving the rest as zeros.

If we rearranged the rows of B^{-1} to be in the order corresponding to the switching order vector $\mathcal{C}$ then we would have a lower triangular matrix which is clearly invertible. Since rearranging the columns does not affect the invertibility of a matrix it follows that B^{-1} is invertible. □

Theorem E.2. *For a system with only one control, the matrix B^{-1} found by the preceding algorithm is the identity matrix.*

Proof. When there is only one control, the vector $\mathcal{C}$ is all ones. Starting at the first row of B^{-1} we set the first entry to one and then move to the next row and column since the first element of $\mathcal{C}$ corresponds to the current (and only) control. Again, we set the entry to one and move to the next row and column for the same reason. This process sets each element of the diagonal to one. Since the remaining entries are zero and B^{-1} is a square matrix, it is the identity matrix.

This result justifies the statement made in Section 5.4 that the matrix α reduces to the vector ξ in the one control case. □

We can find the matrix B by inverting B^{-1}. In practice, computing the inverse is not necessary since we only need to know the product $B\xi$. We can therefore use Matlab's "backslash" operator.

References

[1] Going where no qubit has gone before. `http://www.uwaterloo.ca/profiles/profile.php?id=93`, April 2009. 2

[2] C. Altafini. Controllability of quantum mechanical systems by root space decomposition of $\mathfrak{su}(n)$. *Journal of Mathematical Physics*, 43(5):2051–2062, 2002. 16, 29

[3] M. Athans. The matrix minimum principle. *Information and control*, 11(5/6):592–606, 1968. 33, 35

[4] J. Baillieul. Geometric methods for nonlinear optimal control problems. *Journal of Optimization Theory and Applications*, 25(4):519–548, 1978. 6, 7, 25, 33, 34, 40, 71, 81

[5] U. Boscain, T. Chambrion, and J. Gauthier. On the $k+p$ problem for a three-level quantum system: Optimality implies resonance. *Journal of Dynamical and Control Systems*, 8(4):547–572, 2002. 12, 42

[6] U. Boscain and G. Charlot. Resonance of minimizers for n-level quantum systems with an arbitrary cost. *ESAIM, Control, Optimisation and Calculus of Variations*, 10:593–614, 2004. 12, 14

[7] U. Boscain and Y. Chitour. Time-optimal synthesis for left-invariant control systems on series so (3). *SIAM Journal on Control and Optimization*, 44(1):111–139, 2005. 13

[8] U. Boscain and P. Mason. Time minimal trajectories for two-level quantum systems with drift. *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 3188–3193, Dec. 2005. 12

[9] U. Boscain and P. Mason. Time minimal trajectories for a spin 1/2 particle in a magnetic field. *Journal of Mathematical Physics*, 47(6):062101, 2006. 13

[10] A. Branczyk, P. Mendonca, A. Gilchrist, A. Doherty, and S. Bartlett. Quantum control of a single qubit. *Physical review. A*, 75(1 A), 2007. 11

[11] T. Bröcker and K. Jänich. *Introduction to Differential Topology*. Press Syndicate of the University of Cambridge, 1982. 78

[12] R. W. Brockett. System theory on group manifolds and coset spaces. *SIAM Journal on Control*, 10(2):265–284, 1972. 11

[13] D. Bruss and G. Leuchs. *Lectures on quantum information*. Wiley, 2007. 25

[14] A. Carlini, A. Hosoya, T. Koike, and Y. Okudaira. Time-optimal quantum evolution. *Physical Review Letters*, 96(6):060503, 2006. 12

[15] A. Carlini, A. Hosoya, T. Koike, and Y. Okudaira. Time-optimal unitary operations. *Physical Review A (Atomic, Molecular, and Optical Physics)*, 75(4):042308, 2007. 12

[16] D. Chang and R. Sepulchre. Time-optimal control of a 3-level quantum system and its generalization to an n-level system. *Dynamics of Continuous, Discrete and Impulsive Systems*, 14:575–592, 2007. 12

[17] C. Chevalley. *Theory of Lie Groups*. Princeton University Press, 1946. 20, 22, 23, 29

[18] R. Curtain and H. Zwart. *An Introduction to Infinite-Dimensional Linear Systems Theory*. Springer-Verlag, 1995. 4, 5

[19] M. Curtis. *Matrix Groups*. Springer-Verlag, 1984. 20

[20] D. D'Alessandro. The optimal control problem on SO(4) and its applications to quantum control. *IEEE Transactions on Automatic Control*, 47(1):87–92, 2002. 12, 13, 42

[21] D. D'Alessandro. *Introduction to Quantum Control and Dynamics*. Chapman & Hall/CRC, 2008. 13, 20, 22, 30

[22] D. D'Alessandro and M. Dahleh. *Optimal control of two-level quantum systems*, volume 6. 2000. 12

[23] J. Gallian. *Contemporary Abstract Algebra*. Houghton Mifflin Company, 2006. 77

[24] P. Golovinski. Pontryagin maximum principle for quantum time-optimum problem. In *Physics and Control, 2005. Proceedings. 2005 International Conference*, pages 710–712, Aug. 2005. 33

[25] L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79(2):325–328, Jul 1997.

[26] V. Guillemin and A. Pollack. *Differential Topology*. Prentice-Hall, Inc., 1974. 78

[27] C. Johnson and J. Gibson. Singular solutions in problems of optimal control. *Automatic Control, IEEE Transactions on*, 8(1):4–15, Jan 1963. 43

[28] V. Jurdjevic. *Geometric Control Theory*. Cambridge University Press, 1997. 11, 33, 78, 81

[29] V. Jurdjevic and H. J. Sussmann. Control systems on lie groups. *J. Diff. Eqns.*, 12(2):313–329, 1972. 11, 13, 16, 23, 24, 25, 26, 28, 29, 40, 70

[30] C. Kaya, S. Lucas, and S. Simakov. Computations for bang-bang constrained optimal control using a mathematical programming formulation. *Optimal Control Applications and Methods*, 25(6):295–308, 2004. 56, 59

[31] C. Kaya and J. Noakes. Computations and time-optimal controls. *Optimal control applications and methods*, 17(3):171–185, 1996. 56, 59

[32] C. Kaya and J. Noakes. Computational method for time-optimal switching control. *Journal of Optimization Theory and Applications*, 117(1):69–92, 2003. 56, 59

[33] N. Khaneja, R. Brockett, and S. J. Glaser. Time optimal control in spin systems. *Physical Review A*, 63:1–13, February 2001. 13, 14, 29, 45, 50

[34] N. Khaneja and S. J. Glaser. Cartan decomposition of su(2n) and control of spin systems. *Chemical Physics*, 267(1-3):11–23, June 2001. 13

[35] N. Khaneja and S. J. Glaser. Efficient transfer of coherence through ising spin chains. *Phys. Rev. A*, 66(6):060301, Dec 2002. 13

[36] N. Khaneja, S. J. Glaser, and R. Brockett. Sub-riemannian geometry and time optimal control of three spin systems: Quantum gates and coherence transfer. *Phys. Rev. A*, 65(3):032301, Jan 2002. 13

[37] N. Khaneja, B. Heitmann, A. Spoerl, H. Yuan, T. Schulte-Herbrueggen, and S. Glaser. Quantum gate design metric. *Arxiv preprint quant-ph/0605071*, 2006. 13

[38] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser. Optimal control of coupled spin dynamics: design of nmr pulse sequences by gradient ascent algorithms. *Journal of Magnetic Resonance*, 172(2):296–305, Feb. 2005. 13, 52

[39] N. Khaneja, T. Reiss, B. Luy, and S. J. Glaser. Optimal control of spin dynamics in the presence of relaxation. *Journal of Magnetic Resonance*, 162(2):311–319, June 2003. 52

[40] E. Knill, R. Laflamme, H. Barnum, D. Dalvit, J. Dziarmaga, J. Gubernatis, L. Gurvits, G. Ortiz, L. Viola, and W. Zurek. Quantum information processing - a hands-on primer. *Los Alamos Science*, 27:2–32, 2002. 7

[41] A. N. Korotkov and D. V. Averin. Continuous weak measurement of quantum coherent oscillations. *Phys. Rev. B*, 64(16):165310, Oct 2001. 11

[42] A. J. Krener. The high order maximal principle and its application to singular extremals. *SIAM J. Control and Optimization*, 15(2):256–293, February 1977. 11, 33

[43] S. Lloyd. Almost any quantum logic gate is universal. *Physical Review Letters*, 75(2):346–349, 1995. 19

[44] F. Lowenthal. Uniform finite generation of SU(2). *Canadian Journal of Mathematics*, 24(4):713–727, 1972. 29

[45] S. Lucas and C. Kaya. Switching-time computation for bang-bang control laws. *American Control Conference, 2001. Proceedings of the 2001*, 1:176–181 vol.1, 2001. 56, 59

[46] G. Morris and R. Freeman. Enhancement of nuclear magnetic resonance signals by polarization transfer. *Journal of the American Chemical Society*, 101(3):760–762, 1979. 52

[47] K. Morris. *Introduction to feedback control*. Academic Press, Inc. Orlando, FL, USA, 2000. 16

[48] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000. 3, 6, 7, 19, 25

[49] N. Nielsen, T. Vosegaard, and A. Malmendal. Analytical and numerical tools for experiment design in solid-state NMR spectroscopy. In *Modern Magnetic Resonance*, volume 1, chapter 23, pages 675–683. Springer Netherlands, 2006. 12

[50] P. M. Pardalos and V. Yatsenko. *Optimization and Control of Bilinear Systems; Theory, Algorithms and Applications*. Springer, 2008. 13

[51] C. Pomerance. A tale of two sieves. *Notices Amer. Math. Soc*, 43:1473–1485, 1996. 1

[52] L. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mishchenko. *The Mathematical Theory of Optimal Processes*. John Wiley & Sons, Inc., 1962. 11, 14, 31, 32, 33

[53] V. Ramakrishna, K. L. Flores, H. Rabitz, and R. J. Ober. Quantum control by decompositions of su(2). *Phys. Rev. A*, 62(5):053409, Oct 2000. 13

[54] C. Ryan, C. Negrevergne, M. Laforest, E. Knill, and R. Laflamme. Liquid-state nuclear magnetic resonance as a testbed for developing quantum control methods. *Physical Review A*, 78(1):12328, 2008. 12, 53

[55] A. Sagle and R. Walde. *Introduction to Lie Groups and Lie Algebras*. Academic Press, 1973. 20

[56] H. Samelson. *Notes on Lie algebras*. Springer-Verlag New York, 1990. 22

[57] S. Sastry. *Nonlinear Systems*. Springer-Verlag, 1999. 18

[58] T. Schulte-Herbrüggen, A. Spörl, N. Khaneja, and S. J. Glaser. Optimal control-based efficient synthesis of building blocks of quantum algorithms: A perspective from network complexity towards time complexity. *Physical Review A (Atomic, Molecular, and Optical Physics)*, 72(4):042331, 2005. 13, 52

[59] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, Carnegie Mellon University, August 1994. 53

[60] P. Shor. Algorithms for quantum computation: discrete logarithms andfactoring. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 124–134, 1994. 2

[61] H. Sussmann. The bang-bang problem for certain control systems in $gl(n, r)$. *SIAM J. Control*, 10(3):470–476, 1972. 11

[62] L. Vandersypen, C. Yannoni, I. Chuang, D. Grant, and R. Harris. Liquid state nmr quantum computing. *Encyclopedia of Nuclear Magnetic Resonance*, 9:687–697, 2002. 52

[63] L. M. K. Vandersypen and I. L. Chuang. Nmr techniques for quantum control and computation. *Rev. Mod. Phys.*, 76(4):1037–1069, Jan 2005. 10, 12

[64] F. Warner. *Foundations of Differentiable Manifolds and Lie Groups*. Springer-Verlag, 1983. 23, 78

[65] W. S. Warren, H. Rabitz, and M. Dahleh. Coherent control of quantum dynamics: The dream is alive. *Science*, 259(5101):1581–1589, March 1993. 11

[66] N. Weaver. On the universality of almost every quantum logic gate. *Journal of Mathematical Physics*, 41(1):240–243, January 2000. 19

[67] N. Weaver. Time-optimal control of finite quantum systems. *Journal of Mathematical Physics*, 41:5262, 2000. 12

[68] R. Wu, C. Li, and Y. Wang. Explicitly solvable extremals of time optimal control for 2-level quantum systems. *Physics Letters A*, 295(1):20–24, 2002. 13, 45, 81

[69] H. Yuan, R. Zeier, and N. Khaneja. Elliptic functions and efficient control of ising spin chains with unequal couplings. *Physical Review A (Atomic, Molecular, and Optical Physics)*, 77(3):032340, 2008. 13