Time-Optimal Control of Quantum Systems: Numerical Techniques and Singular Trajectories

by

Tyler Holden

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, 2011

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

As technological advances allow us to peer into and beyond microscopic phenomena, new theoretical developments are necessary to facilitate this exploration. In particular, the potential afforded by utilizing quantum resources promises to dramatically affect scientific research, communications, computation, and material science.

Control theory is the field dedicated to the manipulation of systems, and quantum control theory pertains to the manoeuvring of quantum systems. Due to the inherent sensitivity of quantum ensembles to their environment, time-optimal solutions should be found in order to minimize exposure to external sources. Furthermore, the complexity of the Schrödinger equation in describing the evolution of a unitary operator makes the analytical discovery of time-optimal solutions rare, motivating the development of numerical algorithms.

The seminal result of classical control is the Pontryagin Maximum Principle, which implies that a restriction to bounded control amplitudes admits two classifications of trajectories: bang-bang and singular. Extensions of this result to generalized control problems yield the same classification and hence arise in the study of quantum dynamics. While singular trajectories are often avoided in both classical and quantum literature, a full optimal synthesis requires that we analyze the possibility of their existence.

With this in mind, this treatise will examine the issue of time-optimal quantum control. In particular, we examine the results of existing numerical algorithms, including Gradient Ascent Pulse Engineering and the Kaya-Huneault method. We elaborate on the ideas of the Kaya-Huneault algorithm and present an alternative algorithm that we title the Real-Embedding algorithm. These methods are then compared when used to simulate unitary evolution.

This is followed by a brief examination on the conditions for the existence of singular controls, as well possible ideas and developments in creating geometry based numerical algorithms.

Acknowledgements

I would like to thank my thesis supervisors, Dr. Joseph Emerson and Dr. Kirsten Morris, for their patience and guidance in helping my research. I would also like to express my gratitude to my thesis committee members, Dr. Dong Eui Chang and Dr. Frank Wilhelm, for volunteering their time to oversee my defence.

Completing this thesis would not have been possible without the love and support of my dearest friends: Heather McDonald, who endured countless fits of crankiness and complaining, always replying with a kind word or a slap to the face; Matthew Quinn, who was my moral compass, chief psychiatrist, and proofreader; David Heuser, the sober second opinion and anchor of our group; William Cheng, who showed me my love for the ice; and to all others who played a role in keeping me from imploding from stress and the weight of my own ego.

Finally, I thank my parents for giving me the ambition, motivation, and skill with which to succeed; and my brother, who always stood behind me regardless of how much I hit him.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction and Motivation

As science continues to rapidly develop, so too does our ability to analyze and manipulate the environment in which we inhabit. Whether this growth results in the ability to manoeuvre the incredibly large or the incredibly small, as we push the envelope of the classical description of physics we will need to change our theoretical models. Quantum mechanics is one of those theoretical changes, and is a celebrated theory that describes how microscopic systems interact. It has a rich mathematical background that is amenable to abstract analysis and we will exploit this in our studies.

Recent research into quantum mechanics, especially with regards to entanglement and superposition, has produced insight into physical resources from which radical new technologies may be discovered. Examples of such theoretical research come from the pioneering efforts of Shor[69] and Grover[32] in finding quantum algorithms that yield dramatic improvements over their classical counterparts. Conversely, experimental researchers are pursuing ideas in diverse fields such as nuclear magnetic resonance, super-conducting quantum interference devices, and ion trap quantum computers.

Proposals for applying these new quantum resources have been made in communications, information processing, and nanotechnology. Perhaps the most prominent of these relates to the construction of a quantum computer. With advances in computational hardware following the exponential tendencies of Moore's law, the steady increase in integrated-circuit transistor counts has similarly resulted in the drastic miniaturization of electronic components. Following this trend, fabrication processes will leave engineers unable to ignore the laws of quantum mechanics, as quantum effects will begin to play a radical role in system dynamics. Rather than circumvent these effects, the study of quantum computing hopes to exploit foundational quantum mechanical principles to create a computational

framework that exceeds anything that can be classically obtained.

There are many obstacles that still need to be faced in order for any of these technologies to become a reality. Active research fields include the theoretical description and design of quantum devices, and the physical implementation of quantum systems. The final element is to construct the bridge between these two fields which analyzes how to use physical technologies to implement the abstract procedures described theoretically. This is done by examining how to affect the physical environment in order to establish a prescribed configuration and is known as quantum control theory.

Quantum mechanical systems are intrinsically difficult to control due to their sensitivity to the environment and perturbation. Work with quantum systems would be ideally performed in perfectly isolated environments which are impossible to attain. An interaction between a quantum system and its environment may cause it to lose coherence, becoming classical in nature and forfeiting the advantage afforded by quantum resources. Alternatively, the system may become entangled with the environment in an undesirable way, resulting the unexpected behaviour. One of the greatest challenges in quantum computing is to derive solutions that minimize such interactions.

Traditionally, one can resolve the issue of noisy external systems by implementing a feedback controller. Such a controller measures how the system is varies from its ideal description and adjusts the controls accordingly. Unfortunately, feedback control of a physical system requires an output measurement to determine the appropriate reaction. This very measurement itself represents an interaction with an external system, ironically implying that in order to combat noise, we must induce noise.

It may be more prudent to instead analyze an open-loop control implementation. It has been suggested in [45] that while we cannot avoid the effects of external systems, we can mitigate their influence by minimizing the amount of time allowed for undesirable interactions. This motivates us to examine time-optimal controls in quantum synthesis and is the principle goal of this treatise.

One of the major interests of quantum control theory is the implementation of operators on quantum bits (qubits). Such operators act analogously to circuit gates in classical computing and are often referred to as quantum gates by comparison. Operators on isolated, finite dimensional quantum mechanical systems are represented via unitary matrices and undergo an evolution as directed by the Schrödinger equation. While quantum states also evolve under the Schrödinger equation, the power of using the Heisenberg picture of time evolving operators lies in the ability to disregard state-to-state transitions and instead consider the more general picture of how a system acts on arbitrary preparations.

This thesis will consider all of these results and present our efforts to implement time-optimal controls. The remainder of this thesis is organized as follows: Chapter 3 will present an introduction to the field of quantum control in general, so as to familiarize the reader with active research. This will be followed by a brief review of classical control in Chapter 2 and will include a development of the tools necessary to our research. We follow this with a description of two existing numerical methods given formulated in [37] and [48] in Chapter 4 and a comparison of these methods with those of our own design. We conclude by considering the existence of singular controls and the special role they play in operator synthesis in Chapter 5.

Chapter 2

Introduction to Control Theory

The history of optimal control can trace its roots to the study of variational calculus, and is the manipulation of system dynamics to achieve a desired goal. In particular *optimal* control seeks to construct state trajectories that are the extrema of a functional known as the cost function.

Our goal throughout this section will be to introduce the reader to the general theory and mathematics of optimal control, with a specialization to a time-optimal cost function in a quantum mechanical formalism. We will begin by reviewing some fundamental notions in classical control before extending these results to a more appropriate setting.

2.1 Pontryagin's Principle and Control in $\mathbb{R}^n$

We begin by introducing the general framework of optimal control theory. Much of the definitions here are a combination of the results from Sontag's book [70], Jurdjevic's book [39], and the foundational work of Pontryagin [61].

Definition 2.1.1. Let $n, m \in \mathbb{N}, [t_0, t_1] \subset \mathbb{R}$, $\mathcal{X} \subseteq \mathbb{R}^n$ be an open set and $\mathcal{U} \subseteq \mathbb{R}^m$ be a metric space. Define the function $f : \mathcal{X} \times \mathcal{U} \to \setminus^n$ such that for any fixed $t \in [t_0, t_1], u \in \mathcal{U}$, f is continuously differentiable over all $\mathcal{X}$. Let $x : \mathbb{R} \to \mathbb{R}^n, u : \mathbb{R} \to \mathbb{R}^m$ and consider the differential system defined by

$$\frac{dx(t)}{dt} = f(x(t), u(t)), \qquad x(t_0) = x_0 \in \mathcal{X} \tag{2.1}$$

We say that (2.1) is a *control system* for the control set $\mathcal{U}$ if f is continuously differentiable, its integral curves $x(t)$ are an absolutely continuous function, and $u(t)$ is essentially bounded and measurable. Further, u is said to be admissible if it satisfies all of the above criteria.

Definition 2.1.2. Let x, u, f be as defined in Definition 2.1.1. Take $x_1 \in \mathcal{X}$ and let $f_0 : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$. An *optimal control problem* is to find a control $u(t) \in \mathcal{U}$ such that the evolution of $x(t)$ as determined by (2.1) satisfies $x(t_0) = x_0$ and $x(t_1) = x_1$ and the functional

$$J(x, u) = \int_{t_0}^{t_1} f_0(x(t), u(t))\ dt \tag{2.2}$$

is minimal amongst all such control laws $u(t)$. The system is a *time-optimal control problem* whenever we take $f_0(x(t), u(t)) \equiv 1$.

Controllability is an important topic in the study of control systems. It can be simply stated without worrying about technical details, but is sufficiently fundamental that it should not be overlooked.

Definition 2.1.3. For all $T > 0$ and each $x_0 \in \mathcal{X}$, we define the *accessible or reachable set from x_0 in time T*, $\mathcal{A}(x_0, T)$, as the set of all points in $\mathcal{X}$ for which there exists an admissible control that drives x_0 to that point in time T under the evolution (2.1). In a more mathematically precise sense,

$$\mathcal{A}(x_0, T) = \left\{ x_f \in \mathcal{X} \;\middle|\; \exists u \in \mathcal{U}, x(T) = x_f, \frac{dx(t)}{dt} = f(x(t), u(t)) \right\} \tag{2.3}$$

The *accessible set* $\mathcal{A}(x_0)$ *from* x_0 is the set of all points reachable in arbitrary time T, or more precisely

$$\mathcal{A}(x_0) = \bigcup_{T \geq 0} \mathcal{A}(x_0, T). \tag{2.4}$$

Definition 2.1.4. A control system is completely controllable if $\mathcal{A}(x_0) = \mathcal{X}$ for every $x_0 \in \mathcal{X}$; that is, every point is reachable from every other point.

It will turn out that in the case of quantum mechanical systems the notion of controllability from the viewpoint of accessible sets can be simplified even further. However, we defer that discussion until section 2.4.

Now that we have established the structure of the general problem we must consider the issue of actually finding solutions. While special techniques may be employed for very

specific systems, it would be far better to have an all encompassing theorem from which solutions may be derived. Indeed, the seminal work of Pontryagin [61] is considered by many to be the origin of contemporary optimal control theory, and within this tome is arguably the most fundamental result in the entire field.

Theorem 2.1.5 (Pontryagin Maximum Principle (PMP), Pontryagin [61])**.** *Consider the optimal control problem given by Definition 2.1.2. Furthermore, let $u(t), t_0 \leq t \leq t_1$ be an admissible control such that the corresponding trajectory $x(t)$ begins at the point x_0 at time t_0 and passes through x_1 at time t_1, where the terminal time t_1 is free to vary. In order that $u(t)$ and $x(t)$ be optimal it is necessary that there exist a nonzero continuous vector function $\lambda(t) = (\lambda_0(t), \ldots, \lambda_n(t))$ called the* costate*, such that defining*

$$\mathcal{H}(\lambda, x, u) = \lambda_0 f_0(x, u) + \langle \lambda(t), f(x, u) \rangle \tag{2.5}$$

results in the following conditions:

1. *For every $t \in [t_0, t_1]$ the function $\mathcal{H}(\lambda, x, v)$ in the variable $v \in \mathcal{U}$ attains its maximum at $u(t)$:*
$$\mathcal{H}(\lambda(t), x(t), u(t)) = \sup_{v \in \mathcal{U}} \mathcal{H}(\lambda(t), x(t), v), \tag{2.6}$$
2. *at the terminal time t_1 it is necessary that $\lambda_0(t) \leq 0$ and*
$$\sup_{v \in \mathcal{U}} \mathcal{H}(\lambda(t_1), x(t_1), v) = 0, \tag{2.7}$$
3. *the state and costate satisfy Hamilton's equations*
$$\frac{dx}{dt} = \frac{\partial \mathcal{H}}{\partial \lambda}, \qquad \frac{d\lambda}{dt} = -\frac{\partial \mathcal{H}}{\partial x}. \tag{2.8}$$

Furthermore, it turns out that if $x(t), \lambda(t)$ and $u(t)$ satisfy (2.8) *and Condition 1, the time functions $\lambda_0(t)$ and $H(\lambda(t), x(t), u(t))$ are constant, so that Condition 2 may be verified at any time $t, t_0 \leq t \leq t_1$ and not just at t_1.*

Here, $\langle \cdot, \cdot \rangle$ is the standard Euclidean inner-product on $\mathbb{R}^n$ and $\mathcal{H}(\lambda, x, u)$ defined in (2.5) is referred to as the *Pontryagin Hamiltonian*.

Definition 2.1.6. Consider the optimal control problem given by Definition 2.1.2 and let $u(t) = (u_1(t), \ldots, u_m(t))$ be a coordinate-wise expression for the control variable $u(t) \subseteq \mathcal{U} \subseteq \mathbb{R}^m$. If for each i there exists $m_i, M_i \in \mathbb{R}$ satisfying $-\infty < m_i < M_i < \infty$ such that $\forall t \in [t_0, t_1], m_i \leq u_i(t) \leq M_i, i = 1, \ldots, m$ then we say that the system has bounded controls. Otherwise, we say that the system has unbounded controls.

In the case of bounded controls, we often assume for simplicity that the control are normalized $0 \leq u_i(t) \leq 1$ or symmetrically bounded $-M \leq u_i(t) \leq M$ for some $M > 0$.

For the following argument, assume that the controls have symmetric bounds $|u_i(t)| \leq M_i$. If we choose a vector representation of the control dynamics $f = (f_1, \ldots, f_n)$ and assume that each is affine in the control variable, we can write

$$f_i(x(t), u(t)) = \sum_{j=1}^{m} h_{ij}(x(t))u_j(t) \tag{2.9}$$

for some appropriately chosen functions $h_{ij}(x(t))$. By exploiting the definition of the standard Euclidean inner product, the Pontryagin Hamiltonian will then become

$$\begin{aligned}
\mathcal{H}(\lambda(t), x(t), u(t)) &= \lambda_0 + \sum_{i=1}^{n} \lambda_i(t) f_i(x(t), u(t)) \\
&= \lambda_0 + \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_i(t) h_{ij}(x(t)) u_j(t) \\
&= \lambda_0 + \sum_{j=1}^{m} \left[\sum_{i=1}^{n} \lambda_i(t) h_{ij}(x(t)) \right] u_j(t) \\
\mathcal{H}(\lambda(t), x(t), u(t)) &= \lambda_0 + \sum_{j=1}^{m} \psi_j(t) u_j(t)
\end{aligned} \tag{2.10}$$

where $\psi_j(t) = \sum_{i=1}^{n} \lambda_i(t) h_{ij}(x(t))$. These are referred to as the *switching functions* for a reason that will become clear shortly. Now by applying the PMP and specifically the condition specified by (2.6) we get that an optimal control must maximize (2.10) in the control variable. With bounded control we see that if $\psi_j(t) > 0$ we need to make $u_j(t)$ as large as possible and vice-versa in the case where $\psi_j(t) < 0$. In particular, we get that the control law is governed by $u_j(t) = \text{sgn}(\psi_j(t))M_j$. This leads us to the following definition:

Definition 2.1.7. Consider the time-optimal problem given in Definition 2.1.2 with symmetric, bounded controls. Given that the control law is subject to $u_j(t) = \text{sgn}(\psi_j(t))M_j$ for some appropriately defined switching function $\psi_j(t)$, we say that the control $u_j(t)$ is *bang-bang* if the roots of $\psi_j(t)$ have zero measure. Conversely, if there exists a non-null subset $I \subseteq [t_0, t_1]$ such that $\psi_j(t) \equiv 0$ for almost every $t \in I$ then the control $u_j(t)$ is said to be *singular* on I. If there is a non-null subset $I \subseteq [t_0, t_1]$ on which $\psi_j(t) \equiv 0$ almost everywhere, $j = 1, \ldots, m$ then we say that the controls are *totally singular* on I.

In the case of non-singular bang-bang control laws, a complete knowledge of the switching functions is sufficient to completely resolve the problem. However, we notice that the switching functions depend on the costate $\lambda_i(t)$ which can greatly complicate matters; furthermore, there is no *a priori* way to rule out the existence of singular controls. Singular controls are often neglected in the literature, possibly owing to the fact that such controls are rarely optimal in linear regimes.

There is one more level of classification that is important to affine, bounded-control systems pertaining to the value of λ_0. In the event that $\lambda_0 \neq 0$ we often choose to normalize it so that $\lambda_0 = -1$. However, it may be possible that $\lambda_0 \equiv 0$. The PMP only dictates that the pair $(\lambda_0, \lambda(t))$ not vanish at any point, but it may be possible that $\lambda_0(t)$ is almost everywhere 0.

Definition 2.1.8. Consider the time-optimal problem given in Definition 2.1.2 with bounded controls. In the event that $\lambda_0(t) \equiv 0$ for almost every $t \in [t_0, t_1]$ we say that the controls and corresponding trajectories are *abnormal*, otherwise they are normal.

Working with $\mathbb{R}^n$ however is not always appropriate, necessitating some level of generalization. One such candidate for generalization is the Complex Matrix Maximum Principle (CMMP) which is essentially a restatement of the PMP with matrices in lieu of vectors. The proof of the CMMP is not particularly complex, done by embedding $n \times n$ complex valued matrices into $2n^2$-dimensional real space and then applying the PMP[37]. While we will make little use of this theorem, we include it here for completeness.

Theorem 2.1.9 (Athans [7] and Huneault [37])**.** *Let $M_n(\mathbb{C})$ denote the set of all $n \times n$ matrices with complex valued entries. Consider a dynamical system given by $\frac{d}{dt}X(t) = F(X(t), u(t))$ where $X : [t_0, t_1] \to M_n(\mathbb{C}), u : [t_0, t_1] \to U$ for some admissible control set $U \subseteq \mathbb{R}^n$, and $F : M_n(\mathbb{C}) \times \mathbb{R}^m \to M_n(\mathbb{C})$. Let $t_0, X(t_0)$, and $X(t_1)$ be prescribed quantities with t_1 free to vary, and $u(t)$ be a control that drives $X(t_0)$ to $X(t_1)$. If $F_0 : M_n(\mathbb{C}) \times \mathbb{R}^m \to \mathbb{R}$ is a cost function then for $u(t)$ to minimize $\int_{t_0}^{t_1} F_0(X(t), u(t))\, dt$ it is necessary that there exists $\lambda_0 \in \mathbb{R}$ and a curve $\Lambda : [t_0, t_1] \to M_n(\mathbb{C})$ such that if*

$$H(X(t), \Lambda(t), \lambda_0, u(t)) = \lambda_0 F_0(X(t), u(t)) + \Re \langle \Lambda(t), F(X(t), u(t)) \rangle$$

where $\langle A, B \rangle = \mathrm{Tr}[A^\dagger B]$, then

1. *It is not the case that $\Lambda(t)$ is the zero matrix and λ_0 is zero.*
2. *$(X(t), \Lambda(t))$ evolve according to Hamilton's equations*

$$\frac{d}{dt}X(t) = \frac{\partial H}{\partial \Lambda}(X(t), \Lambda(t), \lambda_0, u(t)), \qquad \frac{d}{dt}\Lambda(t) = -\frac{\partial H}{\partial X}(X(t), \Lambda(t), \lambda_0, u(t))$$

3. $\lambda_0 \leq 0$

4. $u(t) = \underset{v \in U}{\operatorname{argmax}} H(X(t), \Lambda(t), \lambda_0, v)$

This theorem was initially introduced by [7] for the case of matrices over $\mathbb{R}$. It was extended into the complex domain in [37], requiring a greater deal of algebra but keeping to the general spirit of the original proof. We note that this choice of inner-product is not arbitrary. The Hilbert-Schmidt inner product makes $M_n(\mathbb{C})$ as a space isomorphic to a normed vector space. This allows us to embed the matrix space into $\mathbb{C}^n$ via an invertible vectorization argument, apply the traditional PMP, and then translate back into matrix space.

The PMP only gives necessary conditions for optimality, often generating a large set of candidate solutions from which the optimal ones must be picked. There are some limiting cases in which The Maximum Principle also gives sufficient criteria for optimality, though these only occur in real, linear systems [50]. Furthermore, while all smooth, finite-dimensional manifolds can be embedded into $\mathbb{R}^m$ for sufficiently large m, it is not always prudent to examine the properties of those embeddings. Results discovered in one regime can be made extremely complicated by translating across the embedding map, implying that it is often more useful to consider spaces in a natural, embedding free context [19]. In this sense, the CMMP is just an embedding of matrix space into a sufficiently large real space, wherein we have ignored the rich and complicated structure afforded to us had we instead considered the state space as a manifold.

2.2 Geometric Control Theory

We will present here an introduction to some of the basic concepts of geometric control theory that will prove useful in our analysis of geometric control algorithms (Appendix D) and singular controls (Chapter 5).

In the previous section we began by defining a control system in terms of a differential equation. We hope to do the same thing here, but in working with general manifolds we need to be more rigorous and formal. There are many subtle changes that need to be made that are not evident in the case of $\mathbb{R}^n$ or $\mathbb{C}^n$, which occurs as a result of these fields being identical to geometric constructs that do not generalize well.

2.2.1 Differential Geometry

Definition 2.2.1. Let M be a set endowed with a topology. For $n \in \mathbb{N}$ we say that M is a *topological n-manifold* if it is Hausdorff, second countable and locally n-Euclidean. Hausdorff implies that any two distinct points in M can be separated by disjoint open sets. Second countable implies that the there exists a countable basis for the topology on M, and locally Euclidean implies that for any point $p \in M$ there exists U an open neighbourhood of p such that there is a mapping $\phi : U \to \mathbb{R}^n$ which is a homeomorphism onto its image. The collection of all (U, ϕ) as p spans over M is called a (coordinate) *chart* for p.

Topological manifolds appear to be Euclidean spaces if one takes sufficiently small neighbourhoods of points. In order to be able to extend any notion of calculus to these spaces, it will be necessary to define a smooth structure on the manifold. Intuitively, we use coordinate charts (U, ϕ) to move functions on manifolds to functions on $\mathbb{R}$. This allows us to apply standard calculus techniques to these functions before casting them back into a manifold framework. However, for this to work properly we need to ensure that coordinate charts are compatible with one another.

Definition 2.2.2. Let M be a topological space. Two charts (U, ϕ) and (V, ψ) are said to be *smoothly compatible* if whenever $U \cap V \neq \emptyset$, the map

$$\psi \circ \phi^{-1} : \phi(U \cap V) \to \psi(U \cap V) \tag{2.11}$$

is a diffeomorphism. An *atlas* is a collection of charts whose domain covers M, and a *smooth atlas* is an atlas of smoothly compatible charts.

Every smooth atlas can be extended to a unique maximal atlas [19] which is an atlas that is not strictly contained in any other atlas. We define a smooth structure to be a maximal smooth atlas. A smooth manifold is thus a topological manifold endowed with a choice of smooth structure. As mentioned before, these charts are necessary to perform calculus on manifolds.

Definition 2.2.3. Let M, N be two smooth manifolds of dimension m and n respectively, and take $F : M \to N$. For a chart (U, ϕ) of M define $\phi(U) = \tilde{U} \subseteq \mathbb{R}^n$ and similarly for charts of N. We say that F is a *smooth function* between M and N if for any chart (U, ϕ) of M and (V, ψ) of N the function

$$\psi \circ F \circ \phi^{-1} : \tilde{V} \to \tilde{U} \tag{2.12}$$

is smooth.

Fundamental to our study will be the notion of curves on a smooth manifold. These are simply real-parameterizations of paths along the manifold. Alternatively, curves are manifold functions whose domain is a subset of $\mathbb{R}$. We will use these when we solve differential equations on manifolds, so that solutions are actually given by curves.

Definition 2.2.4. Let $I \subseteq \mathbb{R}$ be a connected interval. A *curve* on a smooth manifold M is any continuous mapping $\gamma : I \to M$.

Next we introduce the notion of tangent spaces, for which there are many coordinate-independent constructions. In the following, we will define the derivation definition of tangent vectors. This is motivated by the bijective correspondence between tangent spaces on $\mathbb{R}$ and directional derivatives, that can be seen in Appendix A.1.

Definition 2.2.5. Let M be a smooth manifold. A linear map $X : C^\infty(M) \to \mathbb{R}$ is a *derivation* at the point $p \in M$ if

$$X(fg) = f(p)X(g) + g(p)Xf, \qquad \forall f, g \in C^\infty(M);$$

that is, derivations are linear functionals on smooth functions that satisfy the Leibniz rule. We then define the *tangent space* at $p \in M$ to be

$$T_pM = \left\{ X : C^\infty(M) \to \mathbb{R} \;\middle|\; X \text{ is a derivation} \right\}.$$

Definition 2.2.6. Let M, N be smooth manifolds and $F : M \to N$ be a smooth function. For a fixed $p \in M$ we define the pushforward of F at p as the function $(F_*)_p : T_pM \to T_{F(p)}N$ that acts as

$$(F_*)_p X_p(f) = X_p(f \circ F) \tag{2.13}$$

for $X_p \in T_pM$ and $f \in C^\infty(N)$. Note that this is well defined since $f \circ F : M \to \mathbb{R}$ and $X_p : C^\infty(M) \to \mathbb{R}$.

The above definition of the tangent space is very restrictive since it requires that we specify a point in the manifold. In order to properly consider all possible tangent vectors and ways of moving between tangent spaces, it will be necessary to introduce the fibre bundle.

Definition 2.2.7. Consider three topological spaces E, B and F with a projective mapping $\pi : E \to B$ a continuous surjection. Then

$$\forall x \in E, \exists U \ni \pi(x), \exists f : \pi^{-1}(U) \to U \times F$$

with f a homeomorphism satisfying

$$\text{proj}_1 \circ f = \pi \tag{2.14}$$

where $\text{proj}_1 : U \times F \to U$ is the projection of the Cartesian product onto its first component, then we say that π-admits a *local-trivialization* of E over F and $\pi : E \to B$ is said to be a *fibre bundle*. Alternatively, we will say that E is a fibre-bundle of B over F.

Essentially, this definition is dictating that E appear locally to be the Cartesian product of the base space B with a space F. This is equivalent to locally "gluing" together spaces of F adjoined to points on B, and can be easily visualized in the following definition.

Definition 2.2.8. Let M be a smooth n-manifold and define the *tangent bundle* TM as

$$TM = \bigsqcup_{p \in M} T_p M \tag{2.15}$$

where $\bigsqcup$ denotes the disjoint union. To be more precise, we specify a choice of disjoint union as

$$TM = \bigcup_{p \in M} \{p\} \times \{T_p M\} \tag{2.16}$$

so that tangent vectors are associated to the tangent space in which they lie. TM is a fibre-bundle of M over $\mathbb{R}^n$ and a typical element of TM can be written as (p, X_p) for $p \in M$ and $X_p \in T_p M$. The projection mapping is given by $\pi(p, X) = p$.

Our tangent bundle is the union of all the tangent spaces on M and it can be shown [52] that if M is a smooth n-manifold that TM is a smooth $2n$-manifold. Notice that we can extend our definition of the pushforward to exploit this new structure. If $F : M \to N$ for smooth manifolds M and N, we can then define $F_* : TM \to TN$ where $(F_*)_p : T_p M \to T_{F(p)} N$ is the normal pushforward given in definition 2.2.6. This sort of generalization can be applied to many different tangent space definitions.

The tangent bundle will give us a way of defining vector-fields on M. Since the projection mapping is a mapping between smooth manifolds, we will be able to assign an interpretation to the meaning of a *smooth* vector field. Vector fields will be essential to interpreting and defining the meaning of differential equations.

Definition 2.2.9. Let M be a smooth manifold and $\pi : E \to M$ be a fibre bundle on M. We define a *section* of π to be any function $f : M \to E$ such that $\pi \circ f = \text{id}_M$ where id_M is the identity mapping $\text{id}_M : M \to M, \text{id}_M(p) = p$. We denote the set of all smooth sections by

$$\Gamma(E) = \left\{ f : M \to E \;\middle|\; f \text{ is a smooth section of } E \right\}. \tag{2.17}$$

Definition 2.2.10. A *smooth vector field* X on a smooth manifold M is a smooth section of the tangent-bundle; that is, $X \in \Gamma(TM)$.

If one unravels the definition of a smooth vector field, it is clear that $X \in \Gamma(TM)$ acts precisely as we expect Euclidean vector fields to act. In particular, since $X : M \to TM$ then X takes each point on the manifold to a tangent vector, and small variations around the point $p \in M$ result in smooth variations of the tangent vector. Interestingly, the properties that we have given to vector fields means that $\Gamma(TM)$ can be realized as a $C^\infty(M)$-module [52], a topological space under the C^∞-Whitney topology [22], or as an algebra [39] as will be demonstrated shorty.

Definition 2.2.11. Let M be a smooth manifold. A *Lie bracket* on $\Gamma(TM)$ is a binary operator $[\cdot,\cdot] : \Gamma(TM) \times \Gamma(TM) \to \Gamma(TM)$ defined by $[V,W]f = VWf - WVf$ for all $f \in C^\infty(M)$.

This definition of a Lie bracket is done in a coordinate-free manner, necessitating the use of a function to characterize its action. If instead we were to choose a coordinate basis (x^i) so that we could express $V = V^i \frac{\partial}{\partial x^i}, W = W^i \frac{\partial}{\partial x^i}$ we would get

$$\begin{aligned}
[V,W]f &= V^i \frac{\partial}{\partial x^i}\left(W^j \frac{\partial f}{\partial x^j}\right) - W^j \frac{\partial}{\partial x^j}\left(V^i \frac{\partial f}{\partial x^i}\right) \\
&= V^i \frac{\partial W^j}{\partial x^i}\frac{\partial f}{\partial x^j} + V^i W^j \frac{\partial^2 f}{\partial x^i x^j} - W^j \frac{\partial V^i}{\partial x^j}\frac{\partial f}{\partial x^i} - W^j V^i \frac{\partial^2 f}{\partial x^j x^i} \\
&= V^i \frac{\partial W^j}{\partial x^i}\frac{\partial f}{\partial x^j} - W^j \frac{\partial V^i}{\partial x^j}\frac{\partial f}{\partial x^i},
\end{aligned} \tag{2.18}$$

where in the last line we used the fact that f being smooth results in commuting second derivatives. Since f was arbitrary, we find that $[V,W] = \left(V^i \frac{\partial W^j}{\partial x^i} - W^i \frac{\partial V^j}{\partial x^i}\right)\frac{\partial}{\partial x^j}$ which will be used in Chapter 5.

It can be shown [19] that the Lie bracket of two smooth vector fields is a smooth vector field, assuring us that the Lie bracket is indeed a binary operator on $\Gamma(TM)$. Hence $\Gamma(TM)$ is an algebra under addition and Lie-bracket multiplication.

Now that we have analyzed aspects of the tangent space, it is a natural algebraic question to consider the corresponding dual space of the constructed vector spaces. We recall that for V a finite dimensional real vector space, the set of all (continuous) linear functionals on V forms the (continuous) dual space V^*. More precisely,

$$V^* = \left\{ f : V \to \mathbb{R} \;\middle|\; f \text{ is linear} \right\}.$$

Definition 2.2.12. Let V and W be finite dimensional real vector spaces and $A : V \to W$ a linear map between them. We define the transpose or *pullback* of A as $A^* : W^* \to V^*$ defined by $(A^*\omega)(X) = \omega(AX)$ with $\omega \in W^*$ and $X \in V$.

Since tangent spaces can be realized as vector spaces, we can similarly consider their dual spaces.

Definition 2.2.13. Let M be a smooth n-manifold. For each fixed $p \in M$ we define the cotangent space at p as $T_p^*M = (T_pM)^*$; that is, the cotangent space is the set of all continuous linear functionals on the tangent space. This implies that if $\phi \in T_p^*M$ and $X \in T_pM$ then $\phi(X) \in \mathbb{R}$. Correspondingly, we define the cotangent bundle T^*M as

$$T^*M = \bigsqcup_{p \in M} T_p^*M \tag{2.19}$$

which is a fibre-bundle of M over $\mathbb{R}^n$ with a typical element given by (p, ϕ). The projective mapping is given by $\pi(p, \phi) = p$.

The results that we commented on after defining the tangent bundle also apply to the cotangent bundle. In particular, T^*M is a smooth $2\dim(M)$ dimensional smooth manifold whenever M is smooth, and in this case the projection $\pi : T^*M \to M$ is smooth. We can also define smooth covector fields as smooth sections of π, and denote the set of all smooth covector fields as $\Gamma(T^*M)$.

We have defined the pushforward mapping which takes tangent vectors between tangent spaces as determined by a smooth manifold mapping. With the cotangent bundle defined, we are now in a position to consider the pullback mapping, which maps covectors between cotangent spaces.

Definition 2.2.14. Let $F : M \to N$ be a smooth function on smooth manifolds M and N, and take $p \in M$. The pushforward $F_* : T_pM \to T_{F(p)}N$ yields a transpose mapping $(F_*)^* : T_{F(p)}^*N \to T_p^*M$ which will be denoted by F^* and will be referred to as the pullback. It is defined as

$$(F^*\omega)(X) = \omega(F_*X), \qquad \omega \in T_{F(p)}^*N, X \in T_pM$$

It turns out that elements of the cotangent bundle are the most natural for describing "derivatives" for functions in $C^\infty(M)$. We will go into more detail on why this is the case after the following definition:

Definition 2.2.15. Let $f \in C^\infty(M)$ be a function. We define the *differential 1-form* of f as

$$df_p(X_p) = X_p f. \tag{2.20}$$

The choice of calling df a differential 1-form follows from the generic definition of k-forms given in Appendix A.2.

The determination of whether the tangent or cotangent bundle is the natural space for expressing derivatives comes from whether the function is a curve ($\mathbb{R} \to M$) or the function maps onto the reals ($M \to \mathbb{R}$). Perhaps this is best seen via an example with the functions $f : \mathbb{R} \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}$. By considering f, we see that it is parameterized by a single real variable, in which case the derivative $\frac{df}{dt}$ represents a linear approximation to f in $\mathbb{R}^n$; that is, it represents a vector tangent to the curve $f(t)$ and hence exists as an element of the tangent space. Conversely, when we take the gradient of g,we often write it as

$$\nabla g = \sum_{i=1}^{n} \frac{\partial g}{\partial x^i} \frac{\partial}{\partial x^i} \tag{2.21}$$

though this requires a specification of a coordinate frame. It is easy to show that different coordinates can yield different gradients and so this does not give a good method for describing a generalized derivative on a manifold. We can instead consider the derivative as a one form so that it acts on tangent vectors (which are coordinate independent) hence yielding a coordinate independent representation itself. In the case of real-valued functions, if we are given a vector v we define the 1-form as $dg(v) = (\nabla g) \cdot v$. Hence both the tangent and cotangent spaces are natural for defining derivatives depending on the specifics of the function.

2.2.2 Flow Theory

Flow theory is the consideration of trajectories induced by vector fields. Namely, if X is a local vector field, the goal is to find curves $\gamma : I \to M$ such that $\dot{\gamma}(t) = X_{\gamma(t)}$ for some connected open neighbourhood $I \subseteq \mathbb{R}$. If one restricts this idea to a Euclidean space we see that solving integral curves is equivalent to solving a system of first order differential equations, and will put us in a position to extend the notion of control systems to manifolds. In order to make sense of this equation, we must first evaluate what it means to take the derivative of a path.

Definition 2.2.16. Let $\gamma : [t_0, t_1] \to M$ be a smooth curve on smooth manifold M. Then the tangent vector at $t' \in [t_0, t_1]$ is defined as

$$\dot{\gamma}(t') = \gamma_* \left(\left. \frac{d}{dt} \right|_{t'} \right) \in T_{\gamma(t')} M$$

where $\left. \frac{d}{dt} \right|_{t'}$ is the standard basis vector for $T_{t'}\mathbb{R}$.

Notice that the equality defined by the equation $\dot{\gamma}(t) = V_{\gamma(t)}$ now makes sense as both $\dot{\gamma}(t)$ and $V_{\gamma(t)}$ are elements of the tangent space $T_{\gamma(t)}M$.

Definition 2.2.17. Let M be a smooth manifold and $V \in \Gamma(TM)$. An *integral curve* of V is mapping $\gamma : [a, b] \to M$ such that $\dot{\gamma}(t) = V_{\gamma(t)}, \forall t \in [a, b]$. Additionally, if $0 \in [a, b]$ then we say that $\gamma(0) = p \in M$ is the *initial condition* on γ.

As mentioned earlier, solving integral curves reduces to solving systems of differential equations. This can be seen by fixing an initial condition $p \in M$ and a coordinate representation $(x_i)_{i=1}^n$ in a neighbourhood of p. Assume that $\gamma(t)$ is an integral curve satisfying

$$\dot{\gamma}(t) = V_{\gamma(t)}, \qquad \gamma(0) = p \tag{2.22}$$

for some smooth vector field $V \in \Gamma(TM)$. If we write γ in the coordinate system as $\gamma(t) = (\gamma^1(t), \ldots, \gamma^n(t))$, then (2.22) implies that

$$\dot{\gamma}^i(t) \left. \frac{\partial}{\partial x^i} \right|_{\gamma(t)} = V^i(\gamma(t)) \left. \frac{\partial}{\partial x^i} \right|_{\gamma(t)} \tag{2.23}$$

which, when equating basis coefficients yields the system of differential equations

$$\gamma^i(t) = V^i(\gamma^1(t), \ldots, \gamma^n(t)), \qquad i = 1, \ldots, n. \tag{2.24}$$

We note that by the Existence and Uniqueness Theorem of Differential Equations[19] we are guaranteed that a local solution, and hence a local integral curve, exists. Thus the notion of integral curves is inherently bounded to within neighbourhoods of the initial condition. In the special event that a vector field admits an integral curve for all initial conditions, we give it a special name.

Definition 2.2.18. A vector field $V \in \Gamma(TM)$ is *complete* if $\forall p \in M$ there is a unique integral curve $\theta^p : \mathbb{R} \to M$ of V. We say that θ^p is the flow of V through p.

The initial condition may dramatically change the solution to the integral curve, justifying why we have explicitly included the dependency of θ^p on the point p. By considering a new function that varies with respect to both the initial condition and the time-parameter generated by the integral curve, we get the function $\theta : M \times \mathbb{R} \to M$ that takes $(p, t) \mapsto \theta^p(t)$ where θ^p is defined as in Definition 2.2.18. We refer to θ as the *global flow* of V.

Definition 2.2.19. The vector field associated to a global flow θ is called the *infinitesimal generator* of θ.

This allows us to state the following fundamental theorem which is critical to the study of flows.

Theorem 2.2.20 (Fundamental Theorem of Flows, Lee [52])**.** *Let M be a smooth manifold and $V \in \Gamma(TM)$, not necessarily complete. Take $D \subseteq M \times \mathbb{R}$ to be the maximal subset on which a global flow may be well defined. Then $\exists! \theta : D \to M$ a maximal smooth flow defined on D whose infinitesimal generator is V, and satisfies*

1. *$\forall p \in M, \theta^p : \mathbb{R} \to M$ is the unique maximal integral curve through p.*
2. *$\forall (t, p) \in D, (\theta_t)_* V_p = V_{\theta(t,p)}$; that is, the vector field is invariant under the flow that it generates.*
3. *$\forall t \in \mathbb{R}$ the set $M_t = \left\{ p \in M \,\middle|\, (t, p) \in D \right\}$ is open in M and $\theta_t(p) = \theta(p, t) : M_t \to M_{-t}$ is a diffeomorpism.*

In working with quantum mechanics, we are fortunate that the unitary group has some incredibly powerful properties. One such property is that it is compact, which yields the following useful theorem.

Theorem 2.2.21 (Lee [52])**.** *All smooth vector fields on compact manifolds are complete.*

We will conclude this section with the following definition which allows us to consider the directional derivative of one vector field with respect to another. While it is immensely important in differential geometry and can prove very useful in computational problems, we will consider it primarily as an alternative tool for calculating Lie brackets.

Definition 2.2.22. Let M be a smooth manifold, $V \in \Gamma(TM), p \in M$ and let θ denote the flow of V. We define the vector $(\mathcal{L}_V W)_p$ called the Lie derivative of W with respect to V at p by

$$(\mathcal{L}_V W)_p = \left. \frac{d}{dt} \right|_{t=0} (\theta_{-t})_* W_{\theta_t(p)} = \lim_{t \to 0} \frac{(\theta_{-t})_* W_{\theta_t(p)} - W_p}{t}. \tag{2.25}$$

This definition can be quite cumbersome, and so we present the following theorem that will make computations far more simple.

Theorem 2.2.23. *If $V, W, X \in \Gamma(TM)$ and $f \in C^\infty(M)$ then $(\mathcal{L}_V W) \in \Gamma(TM)$ and*

1. $\mathcal{L}_V W = [V, W]$.
2. $\mathcal{L}_V f = Vf$
3. $\mathcal{L}_V W = -\mathcal{L}_W V$
4. $\mathcal{L}_V [W, X] = [\mathcal{L}_V W, X] + [W, \mathcal{L}_V X]$
5. $\mathcal{L}_{[V,W]} X = \mathcal{L}_V \mathcal{L}_W X - \mathcal{L}_W \mathcal{L}_V X$
6. $\mathcal{L}_V (fW) = (Vf)W + f\mathcal{L}_V W$

2.2.3 Specialization to Lie Groups

As mentioned previously, working in a quantum mechanical atmosphere affords many powerful tools. Another such property is that the unitary group is indeed a Lie group, a special kind of group on which a great deal of research has been performed.

Definition 2.2.24 (Baker [8])**.** Let G be a smooth manifold which is also a topological group with multiplication map mult : $G \times G \to G$ and inverse map inv : $G \to G$ and take $G \times G$ to be the product manifold. Then G is a *Lie group* if mult and inv are smooth maps.

There are three actions of the group on itself that play a crucial role in Lie theory. In particular, they are

$L_g : G \to G$	$L_g(x) = gx$	Left Translation
$R_g : G \to G$	$R_g(x) = xg$	Right Translation
$\chi_g : G \to G$	$\chi_g(x) = gxg^{-1}$	Conjugation

Ultimately, the set of right- and left-invariant vector fields (Definition 2.2.25) will equivalently and symmetrically define what is called a Lie algebra, while the conjugation map will give us the adjoint representations (Definition 2.2.33).

Definition 2.2.25. A vector field $X \in \Gamma(TM)$ is *left-invariant* if it is invariant under the action of $L_g, \forall g \in G$. That is, it is L_g-related to itself

$$(L_g)_* X_{g'} = X_{gg'}, \qquad (L_g)_* X = X.$$

Proposition 2.2.26 (Sastry [66])**.** *The Lie-bracket of two left-invariant vector fields is also a left-invariant vector field.*

Since $(L_g)_*$ is a linear operator, the set of all left-invariant vector fields is a linear subspace of $\Gamma(TM)$. Furthermore, since the Lie-bracket preserves left-invariance we can view the set of all left-invariant vector fields as an algebra.

Definition 2.2.27. A Lie algebra is a couple $(\mathfrak{g}, [\cdot,\cdot])$ with $[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ a binary operator that is bilinear, antisymmetric, and satisfies the Jacobi identity

$$[V,[W,X]] + [W,[X,V]] + [X,[V,W]] = 0, \qquad \forall V, W, X \in \mathfrak{g}. \tag{2.26}$$

From our previous discussion and this definition of a Lie-algebra, it is then easy to see that the left-invariant vector fields form a Lie algebra. Lie algebras can be studied in their own right, though they originally arose in the study of Lie groups [28]. When the group is explicitly specified we say that this set is *the* Lie-algebra and write $\mathrm{Lie}(G)$. We further note that our specification of left-invariance can be switched to right-invariance and the same arguments still hold. In fact, the two Lie-algebras generated are anti-isomorphic, as shown in Appendix B.2. Consequently, there is no loss of generality by considering one construction over the other. In a remarkable and well known theorem, the relationship between a Lie group and its Lie algebra can be concretely stated as follows:

Theorem 2.2.28 (Lee [52])**.** *Let G be a Lie group with identity $e \in G$. The evaluation map $\varepsilon : \mathrm{Lie}(G) \to T_eG$ given by $\varepsilon(X) = X_e$ is a vector space isomorphism. In particular,* $\mathrm{Lie}(G)$ *is a finite dimensional vector space,* $\dim \mathrm{Lie}(G) = \dim(G)$ *and* $\mathrm{Lie}(G) \cong T_eG$.

Let us take a moment to consider the implications of this theorem. For our purposes, let us consider the Lie group generated by the left-invariant vector fields of the group G. By definition, left-invariance allows us to translate a tangent vector from one tangent space to any other tangent space. Since the tangent space at the group identity is isomorphic to the Lie algebra, knowledge of the Lie algebra corresponds to knowledge of the tangent space at any point in G. More precisely, if $g \in G$ then $T_gG \cong (L_g)_* T_eG$.

We next move on to an examination of how flow theory can be specialized in the formalism of Lie groups.

Definition 2.2.29. Let G be a Lie group. A one-parameter subgroup of G is a Lie-group homomorphism $F : \mathbb{R} \to G$. More information on group homomorphisms are given in Appendix B.1.

Note that the one-parameter subgroup is the mapping itself, and hence is not truly a subgroup of G. However, the choice of nomenclature is still reasonable as it can be shown that the image of one-parameter subgroups is subgroup, the proof for which is given in Appendix B.2.

Theorem 2.2.30 (Lee [52])**.** *Let G be a Lie group. The one-parameter subgroups of G are precisely (bijectively) the integral curves of left-invariant vector fields at the group identity. In particular, a one-parameter subgroup is uniquely determined by its initial tangent vector $X \in T_eG$.*

Since all integral curves, and hence flows, are one-parameter subgroups, this theorem implies that they are all generated by elements of the Lie-algebra. A natural question arises regarding the movement between an element of a Lie algebra and the corresponding group. The previous theorem regarding the relationship between integral curves and the Lie algebra will be crucial to defining the desired transformation between the group and its algebra.

Definition 2.2.31. Given a Lie group G with identity $e \in G$, consider its Lie algebra $\mathfrak{g} = \mathrm{Lie}(G)$. We define the *exponential mapping* $\exp : \mathfrak{g} \to G$ which acts as $\exp X = F(e)$ where F is the one parameter subgroup generated by X.

Proposition 2.2.32 (Lee [52])**.** *Let G be a Lie group with Lie algebra $\mathfrak{g}$. Then*

1. *The exponential map $\exp : \mathfrak{g} \to G$ is smooth.*
2. *$\forall X \in \mathfrak{g}$ the mapping $F(t) = \exp tX$ is the one-parameter subgroup generated by X.*
3. *$\forall X \in \mathfrak{g}$ we have that $\exp(s+t)X = \exp sX \exp tX$.*

The exponential map now gives us an explicit method of combining all of the abstract details mentioned earlier. We see that it is the exponential map which creates the flow of the left-invariant vector fields at identity, which are member of the Lie-algebra.

We shall conclude this section with a mention of some representation theory which are important from a theoretical standpoint.

Definition 2.2.33. Let G be a Lie group and denote by $\Psi : G \to \mathrm{Inn}(G)$ the operator that takes g to its corresponding inner-automorphism; that is, $\Psi(g) = \chi_g$ where $\chi_g(h) = ghg^{-1}$ is the conjugation group action we alluded to earlier. Consider the pushforward of this map at the group identity $e \in G$, noting that $\Psi_g(e) = \chi_g(e) = e$ so that $d(\Psi_g)_e : T_eG \to T_eG$.

Since $T_eG \cong \mathrm{Lie}(G)$ we choose to move to a Lie-algebraic framework by defining the map $\mathrm{Ad}_g = d(\Psi_g)_e : \mathrm{Lie}(G) \to \mathrm{Lie}(G)$. Taken as a mapping over all of G we then define the adjoint representation as

$$\mathrm{Ad} : G \to \mathrm{Aut}(\mathfrak{g}), \qquad g \mapsto \mathrm{Ad}_g$$

There is a very similar and related representation on Lie algebras. It is given as follows:

Definition 2.2.34. Given a Lie algebra $\mathfrak{g}$ and $X \in \mathfrak{g}$ we define the adjoint representation of $\mathfrak{g}$ as the endomorphism $\mathrm{ad}_X : \mathfrak{g} \to \mathfrak{g}$ such that $\mathrm{ad}_X(Y) = [X, Y]$ with the usual Lie bracket. Considered as a mapping over all of $\mathfrak{g}$ we then have that $\mathrm{ad} : \mathfrak{g} \to \mathrm{End}(\mathfrak{g})$.

These choices of representation and their correspondingly related notation are not an accident. It turns out that the adjoint representations of groups and algebras arise naturally and frequently in the study of Lie theory, and the relationship between the two is demonstrated in the following two propositions:

Proposition 2.2.35 (Fulton [30] and Sastry [66])**.** *Let G be a Lie group and $\mathfrak{g}$ its Lie algebra. Then for any $X \in \mathfrak{g}$ we have $\mathrm{Ad}_{\exp X} Y = \exp(\mathrm{ad}_X) Y$.*

Proposition 2.2.36 (Fulton [30])**.** *The mapping* Ad *and* ad *are related via* $\mathrm{ad} = \mathrm{Ad}_* = d\,\mathrm{Ad}$.

2.2.4 Geometric Control Theory

In order to properly introduce the previous concepts, we have been forced to maintain a high level of abstraction. However, in order to perform computations it will be necessary to examine what some of these constructs look like if we fix a particular coordinate system. Let $\{x_i\}_{i=1}^n$ be a smooth, local coordinate representation in a neighbourhood of $p \in \mathbb{R}^n$. It is shown in Appendix A.1 that $T_p\mathbb{R}^n \cong \mathbb{R}^n$ and so the partial derivative operator $\frac{\partial}{\partial x^i}\big|_p$ is in fact a derivation at p. By considering the full spectrum of such operators $\left\{ \frac{\partial}{\partial x^i}\big|_p \right\}_{i=1}^n$, we can create a basis for $T_p\mathbb{R}^n$ for any choice of p. Since the motivation for much of the definitions used in differential geometry is to transfer properties of $\mathbb{R}$ to manifolds via coordinate charts, we can use this basis to define a basis for tangent spaces on the manifold.

Definition 2.2.37. Let M be a smooth manifold, $p \in M$, and (U, φ) be a coordinate chart for p. The pushforward of φ^{-1} is $(\varphi^{-1})_* : T_{\varphi(p)}\mathbb{R}^n \to T_p\mathbb{R}^n$ and so gives us a way to push basis vectors from $\mathbb{R}^n$ onto M. Define the *tangent basis vectors* as

$$\left.\frac{\partial}{\partial x^i}\right|_p = (\varphi^{-1})_* \left.\frac{\partial}{\partial x^i}\right|_{\varphi(p)}. \tag{2.27}$$

The function $(\phi^{-1})_* : \mathbb{R}^n \to T_pM$ is a linear map since pushforwards are always linear, and invertible since ϕ is a diffeomorphism. This makes $(\phi^{-1})_*$ a basis preserving bijection and so our choice of definition for the tangent vectors of T_pM guarantees that $\left\{ \frac{\partial}{\partial x^i}\Big|_p \right\}$ defines a basis set for T_pM. We can compute the action of these basis vectors on elements of $C^\infty(M)$ by

$$\frac{\partial}{\partial x^i}\Big|_p f = (\varphi^{-1})_* \frac{\partial}{\partial x^i}\Big|_{\varphi(p)} f \tag{2.28}$$

$$= \frac{\partial}{\partial x^i}\Big|_{\varphi(p)} (f \circ \varphi^{-1}). \tag{2.29}$$

Given any basis $\{E_i\}_{i=1}^n$ for T_pM we can define a dual basis $\{\varepsilon^i\}_{i=1}^n$ on T_p^*M by demanding that

$$\varepsilon^i(E_j) = \delta^i_j \tag{2.30}$$

where δ^i_j is the Kronecker delta function. In the special case where we have prescribed a coordinate basis $\{x^i\}_{i=1}^n$ at $p \in M$ we define the dual basis to $\left\{ \frac{\partial}{\partial x^i}\Big|_p \right\}$ as $\{dx^i|_p\}_{i=1}^n$ where $dx^i|_p \left(\frac{\partial}{\partial x^j}\big|_p \right) = \delta^i_j$.

We are now in a position to give an example as to why the real case of the PMP (Theorem 2.1.5) can subtly hide the geometric details of the system. When we introduce the general Pontryagin Principle (Theorem 2.3.7), we will show that the costate variable is actually a smooth covector field in $\Gamma(T^*M)$ mapping $G \to T^*G$, acting as a linear functional on the smooth vector field that defines the system dynamics. This is not evident in the real case because $\mathbb{R}$ is algebraically self-dual. More precisely, $T^*\mathbb{R}^n \cong T\mathbb{R}^n \cong \mathbb{R}^{2n}$ and so one cannot tell the difference between elements of the tangent, cotangent, and system space.

With the tools of integrals curves, flows, and differential geometry, we can generalize our notion of control systems given in (2.1). In particular, we have the following:

Definition 2.2.38 (Jurdjevic [39])**.** Let G be a smooth manifold and $m \in \mathbb{N}$ with $\mathcal{U} \subseteq \mathbb{R}^m$ a metric space. Consider a mapping given by $F : G \times \mathcal{U} \to TG$ such that for every fixed $u \in \mathcal{U}$ the map $F(\cdot, u)$ is a smooth vector field on G. If $u : G \times \mathbb{R} \to \mathcal{U}$ is an essentially bounded and measurable function, then the the time-varying differential system

$$\frac{dX}{dt} = F(X, u(X, t)) \tag{2.31}$$

is a *control system* on G.

This definition is still very general and is not yet amenable to the structure of integral curves. In particular, we cannot say that F determines a vector field, since its value depends on both the control $u(X,t)$ and the state $X \in G$. In order to fix this caveat, we consider the vector fields generated by fixing a particular control and then consider all possible controls. Hence we define a *family of vector fields* as the set $\mathcal{D} = \left\{ F_u \;\middle|\; u \in \mathcal{U} \right\}$ where $F_u(X) = F(X,u)$. Certainly the set $\mathcal{D}$ now represents every possible admissible vector field as determined by our control set.

Definition 2.2.39. Consider a fibre bundle $\pi : E \to B$ with a local trivialization of E over F. If F is a vector space then we say that $\pi : E \to B$ is a *vector bundle*. In particular, if $\dim_{\mathbb{R}} F = k$ then $\pi : E \to B$ is a *rank-k vector bundle*.

Via this definition, if G is a n-manifold the tangent bundle is a rank-n vector bundle. Since our set $\mathcal{D}$ consists of vector fields, it is a subbundle of the tangent bundle TG. It is possible that $\mathcal{D}$ changes in rank as it varies over G, as different neighbourhoods of G might give rise to different linear independence relations on the elements $F_u \in \mathcal{D}$. In this case it is hard to describe the general geometric structure that $\mathcal{D}$ imposes on the manifold since we must specify the rank of the the vector field family. However, for our purposes we will assume that $\operatorname{rank} \mathcal{D}$ is constant everywhere so that it defines a distribution on G. This will allow us to use the tools of sub-Riemannian geometry to aid our study. Alternatively, if we define $\mathcal{D}|_X = \left\{ F_u(X) \;\middle|\; u \in \mathcal{U} \right\}$ the set of vector fields at X, then we can write (2.31) as a differential inclusion

$$\frac{dX}{dt} \in \mathcal{D}|_X. \tag{2.32}$$

Recall the definition of reachable sets given in Definition 2.1.3. By assuming that the vector fields in $\mathcal{D}$ are all complete, we can give an explicit structure to $\mathcal{A}(X_0)$. In fact, each $X \in \mathcal{A}(X_0)$ can be written as [39]

$$X = \left(\prod_{i=1}^{n_X} \exp\left[t_i H_i \right] \right) X_0, \qquad n_X \in \mathbb{N},\ t_i \in \mathbb{R},\ H_i \in \mathcal{D} \tag{2.33}$$

where n_X depends on the choice of X. We can also use the distribution to characterize the controllability of the system.

Definition 2.2.40. Let $\mathcal{F}, \mathcal{G}$ be distributions on a smooth manifold G. Define the Lie bracket on these distributions as

$$[\mathcal{F}, \mathcal{G}] = \operatorname{span} \left\{ [f,g] \;\middle|\; f \in \mathcal{F}, g \in \mathcal{G} \right\}. \tag{2.34}$$

Furthermore, take $\mathcal{F}^1 = \mathcal{F}$ so that we can recursively define $\mathcal{F}^{i+1} = \mathcal{F}^i + [\mathcal{F}, \mathcal{F}^i]$. By doing this we generate a flag of subsheaves on TG [55]

$$\mathcal{F}^1 \subseteq \mathcal{F}^2 \subseteq \mathcal{F}^3 \subseteq \cdots \subseteq TG. \tag{2.35}$$

If these subsheaves satisfy the ascending chain condition and eventually generate the entire bundle, we say that $\mathcal{F}$ is *bracket generating*. More precisely, $\mathcal{F}$ is bracket generating if $\exists k \in \mathbb{N}$ such that $\mathcal{F}^{k+1} = \mathcal{F}^k = TG$.

Equivalently, we see that if a distribution $\mathcal{D}$ is bracket generating then by fixing a local frame $\{H_i\}$ for $\mathcal{D}$ at $p \in G$, one finds that every element of TG in a neighbourhood of p lies in the linear span of some collection of elements from the set of all iterated Lie brackets of $\{H_i\}$. This is a generalization of *Hörmander's condition* in $\mathbb{R}^n$ to the case of smooth manifolds. The next theorem will allows us to use the notion of bracket generating distributions to reflect on the controllability of such systems.

Theorem 2.2.41 (Chow's Theorem, Calin [20])**.** *Let G be a connected manifold and $\mathcal{D} \subseteq TG$ be a bracket generating distribution. Then any two points of G can be connected by a horizontal curve; that is, a curve whose tangent lies in $\mathcal{D}$ at every point.*

If we consider the problem of minimizing a cost function $f_0(x(t), u(t))$ via $\int_0^T f_0(x(t), u(t))\, dt$ then we can entirely recast an optimal control problem as a sub-Riemannian problem.

Definition 2.2.42. A *generalized, sub-Riemannian, optimal control problem* (GSOCP) is a couple $(f, \mathcal{F})$ where f is an objective function and $\mathcal{F}$ is a horizontal distribution. The goal is then to transfer some point $x_0 \in M$ to another point $x_f \in M$ via a horizontal curve $x(t)$ such that $\dot{x}(t) \in \mathcal{F}, \forall t \in [0, T]$ and $\int_0^T f_0(x(t), u(t))\, dt$ is minimal amongst all horizontal paths.

We have mentioned previously that from a sub-Riemannian point of view, our differential equation (2.31) could instead be viewed as a *differential inclusion*. It is then a simple application of Chow's theorem to say that if our control fields are bracket generating as a distribution and our manifold is connected, our system is controllable.

Next we want to consider the difference between holonomic and non-holonomic systems. Intuitively, let us consider a ball on a flat table which rolls without slipping or twisting. If we consider the entire configuration space of both the ball and the table, we can represent our space as the product manifold $M = \mathbb{R}^2 \times S^2$, which is four dimensional. Note that our distribution is only two dimensional by our restriction to non-slipping and non-twisting

movement. Consider any two configurations in M. Our question is whether or not we can drive one configuration to another only using rolling. Alternatively, is there a horizontal path on $\mathbb{R}^2 \times S^2$ that connects any two points? It turns out that the answer is affirmative [38], and the reason for this is non-holonomy.

Definition 2.2.43. Let $\mathcal{D}$ be a distribution. We say that $\mathcal{D}$ is *holonomic* if $\mathcal{D}$ is integrable, and *non-holonomic* otherwise.

Historically, the reason why we term the constraint set as holonomic is that if a distribution was defined as the kernel of a one-form then that one form could be written in terms of holonomic functions. However, there is a more intuitive way to visualize the notion of holonomy. From our rolling ball example, choose an initial configuration in $\mathbb{R}^2 \times S^2$. Given a fixed perspective, it is simple to see that the action of rolling in the ball "forward" then "right" leaves the sphere in a different configuration than if we had rolled the ball "right" then "forward." This is because the tangent vectors that define rolling in these two independent directions do not commute. This non-commutativity allows us to generate every possibly configuration, and is made mathematically precise by the Frobenius Theorem [52].

2.2.5 Translation Invariant Systems

We will see shortly that the evolution of closed quantum systems is privy to right-invariance. Such translation invariant systems are well studied in dynamics and we will briefly introduce some pertinent points here. We note immediately that the use of right or left in all of the following information is essentially arbitrary and can easily be interchanged.

Recall that if R_X is the group action of right-translation on G, then we define a right-invariant vector field (Definition 2.2.25) $F \in \Gamma(TG)$ to be one in which

$$(R_X)_* F = F, \qquad \text{or alternatively} \qquad (R_X)_* F_Y = F_{YX}, \forall Y \in G.$$

In essence, a right-invariant vector field simply means that right-translation by a group element simply results in the corresponding vector field at that element. We recall that the left-invariant vector fields define a Lie algebra, and since they are isomorphic to the right-invariant vector fields the Lie algebra can be defined in terms of either translation. This might cause one to wonder what the implications of considering a dynamical system under invariant vector fields might be.

Let $F \in \Gamma(TG)$ be a right invariant vector field, or equivalently let $F_0 = F(\mathrm{id}) \in \mathfrak{g}$. Let $\exp[tF_0]$ be the flow of F with initial condition at the identity id. Then $\forall X \in G$ if we

define $X(t) = \exp[tF_0]X$ we get that

$$\frac{dX}{dt} = F_0 \exp[tF_0]X = F_0 X(t), \qquad X(0) = X. \tag{2.36}$$

This motivates the following definition:

Definition 2.2.44. Let G be a Lie group and consider a differential system given by

$$\frac{d}{dt}X(t) = F(X(t)) \tag{2.37}$$

for some some smooth vector field $F \in \Gamma(TG)$. We say that (2.37) is a *right-invariant system* if $\forall Y \in G$ the curve $Y(t) = X(t)Y$ also satisfies (2.37).

We would immediately like to point out an important difference between (2.36) and (2.37). Namely, the former is the product of a Lie-algebraic element with the state, while the latter is a vector field evaluated at the state. In the case of matrix Lie groups, the relationship between these two can be made explicit by considering the following lemma:

Lemma 2.2.45. *Let G be a matrix Lie group of dimension n. Then the pushforward of the right translation operator $R_X(Y) = YX$ acts on tangent vectors $F \in T_{\mathrm{id}}G$ as*

$$(R_X)_*(F) = FX. \tag{2.38}$$

Before we can prove this, we will need another lemma, stated as follows:

Lemma 2.2.46 (Lee [52])**.** *Let $F : M \to N$ be a smooth map and let $\gamma : J \to M$ be a smooth curve. For any $t_0 \in J$, it follows that*

$$\left.\frac{d}{dt}\right|_{t_0} (F \circ \gamma) = F_* \left(\frac{d\gamma}{dt}(t_0)\right) \tag{2.39}$$

Proof. By applying the definition of tangent vectors, we get

$$\left.\frac{d}{dt}\right|_{t=t_0} (F \circ \gamma) = (F \circ \gamma)_* \left.\frac{d}{dt}\right|_{t_0} = F_*\gamma_* \left.\frac{d}{dt}\right|_{t_0} = F_* \left(\frac{d\gamma}{dt}(t_0)\right)$$

□

Proof of Lemma 2.2.45. Let $R_X : M \to M$ be the right-translation group action, and $\gamma : J \to M$ be any smooth curve such that $\gamma(t_0) = \mathrm{id}$ and $\gamma'(t_0) = F$. Via Lemma 2.2.46 it follows that we can calculate $(R_X)_*(F)$ by computing $(R_X \circ \gamma)'(t_0)$. This is done as follows:

$$\begin{aligned}\left.\frac{d}{dt}\right|_{t_0} (R_x \circ \gamma) &= \left.\frac{d}{dt}\right|_{t_0} \gamma(t)X \\ &= \gamma'(t_0)X \\ &= FX.\end{aligned}$$

And so $(R_X)_*(F) = FX$ as required.

□

Now let $F_0 \in \mathrm{Lie}(G)$ be some Lie-algebraic element, and define $F(X(t)) = (R_X)_*F$. By substituting this into (2.37) and using (2.38) we get precisely the expression $\frac{dX}{dt} = F_0X(t)$ as described by (2.36). We would also like to state how crucially important it is that F_0 be a Lie-algebraic element. In particular, since F_0 is then guaranteed to be right-invariant it follows that $F_0X(t) = (R_{X(t)})_*F \in T_{X(t)}G$ which keeps the equation consistent.

Now we need to be somewhat careful in translating our notion of right-invariant systems to a control setting. In particular, the vector field is usually expressed as $F(X, u(X, t))$ and consequently is not a vector field, but instead represents a family of vector fields.

Definition 2.2.47. Let G be a Lie group and $U \subseteq \mathbb{R}^m$ a set of admissible controls. Consider a differential control system given by

$$\frac{d}{dt}X(t) = F(X(t), u(X(t), t)) \tag{2.40}$$

for some smooth function $F : G \times U \to TG$. We say that (2.40) is a *right-invariant control system* if $\forall Y \in G, dR_YF(X, u) = F(R_YX, u)$.

Notice that this definition encapsulates the same information prescribed in Definition 2.2.44 but accounts for the existence of the control functions. In such right-invariant control systems, the notion of controllability through accessible sets becomes very simple. Indeed, if we take $X_0 \in G \setminus \{\mathrm{id}\}$ and the differential system is given by (2.31)

$$\frac{dX}{dt} = F(X, u(X, t)), \qquad X(t_0) = \mathrm{id}$$

then defining $Y(t) = X(t)X_0$ means that $Y(t)$ satisfies

$$\frac{dY}{dt} = F(Y, u(Y,t)), \qquad Y(t_0) = X_0 \tag{2.41}$$

and the accessible set through X_0 can be written as $\mathcal{A}(X_0) = \mathcal{A}(\mathrm{id})X_0$. Since X_0 was arbitrary, controllability for every element of the group can be determined by analyzing the controllability from the identity.

We have examined the very basics of geometric control theory in this section, hoping to provide a more rigorous view of the topics introduced in section 2.1. With these techniques in mind we are now more comfortable with extending control systems to general manifolds using geometric techniques. Our next ambition will be to introduce the Maximum Principle in this manifold framework so that it can facilitate our study in quantum control.

2.3 The Generalized Pontryagin Principle

To consider the extension of the PMP to general manifolds, we will need to consider the natural symplectic nature of the cotangent bundle. Borrowing from the mathematical development of classical mechanics, we take the state space and "lift" the dynamical equations to a state-momentum or phase-configuration space. The momentum operator is given by a linear functional and hence the combined state lives in the cotangent bundle.

Definition 2.3.1. Let G be a smooth manifold and $\Lambda^2 G$ be the set of alternating two tensors, of which smooth sections are differential 2-forms. Then $\omega \in \Lambda^2 G$ is *closed* if $d\omega = 0$, and *non-degenerate* if $\forall \tilde{F} \in \Gamma(TG)$, $\omega(F, \tilde{F}) = 0$ implies that F is the zero section. A *symplectic structure* is a closed, non-degenerate differential 2-form, and a *symplectic manifold* is a manifold with a symplectic structure.

Analogous to the construction of intrinsic Riemannian structures on the tangent bundles of smooth manifolds, the cotangent bundle has a natural symplectic structure.

Definition 2.3.2. Let G be a smooth manifold. We define the *canonical* 1*-form* at $(X, \lambda) \in T^*G$ by $\tau_{(X,\lambda)} = \pi^*\lambda$ where $\pi : T^*G \to G$ is the canonical fibre projection. Since this is a one form, it maps (X, λ) to a covariant 1-tensor which is a linear functional on $T(T^*G)$. If $\zeta \in T(T^*M)$ then

$$\tau_{(X,\lambda)}(\zeta) = \pi^*\lambda(\zeta) = \lambda(\pi_*\zeta). \tag{2.42}$$

Proposition 2.3.3 (Bonnard [12]). *Let G be a smooth manifold. Define $\omega = -d\tau$ (there is no canonical value for the sign) where τ is the canonical 1-form on T^*G. As defined, ω is a symplectic form on T^*G.*

There is a much celebrated result by Darboux which gives us a local expression for ω in terms of local coordinates.

Theorem 2.3.4 (Darboux). *Let (G, ω) be a $2n$-dimensional symplectic manifold. Then $\forall X \in G, \exists (x^1, y_1, \ldots, x^n, y_n)$ smooth coordinates such that in a neighbourhood around X the symplectic form has the structure*

$$\omega = \sum_{i=1}^{n} dx^i \wedge dy_i$$

In particular, the "state-momentum" representation (X, λ) on the cotangent bundle implies that locally we can write $\omega = \sum_{i=1}^{n} dX^i \wedge d\lambda_i$. Our next challenge will then be translating the problem of finding integral curves for $\dot{X}(t) = F(X(t))$ to one that is consistent on the cotangent bundle. This is done as follows:

Definition 2.3.5. Let G be a smooth manifold and consider a smooth vector field $F \in \Gamma(TG)$. We can uniquely lift this vector field to a function on the cotangent bundle, called the *Hamiltonian of F*, and denote it by $\mathcal{H}_F : T^*G \to \mathbb{R}$. If $\xi = (X, \lambda) \in T^*G$ then the action of this Hamiltonian is defined as $\mathcal{H}_F(\xi) = \lambda(F(X))$.

Definition 2.3.6. Let G be a smooth manifold and $\mathcal{H} : G \to \mathbb{R}$ be a smooth function. Then we can uniquely lift this function to a vector field on G, called the *Hamiltonian vector field*. This is done by associating it with the unique vector field $\overrightarrow{\mathcal{H}} \in \Gamma(TG)$ satisfying $\iota_{\overrightarrow{\mathcal{H}}}\omega = -d\mathcal{H}$.

Notice that we can combine these two definitions to lift vector fields on a manifold G, to vector fields on the cotangent bundle T^*G. In particular, if $F \in \Gamma(TG)$ then we define the Hamiltonian of F as $\mathcal{H}_F \in C^\infty(T^*G)$, then lift this to the vector field $\overrightarrow{\mathcal{H}_F} \in \Gamma(T(T^*G))$. We say that $\overrightarrow{\mathcal{H}_F}$ is the *Hamiltonian lift* of F.

This notion of Hamiltonian lift will allow us to consider the more natural setting of "state-momentum space," on which the solution to our optimal control problem will be projections of lifted trajectories. However, the definition above is very obscure and it may

not be immediately evident why it would give the desired results. By discovering the form of the Hamiltonian vector field in terms of the Darboux coordinates, we will find that Hamilton's equations are satisfied. Let us specify some arbitrary point in T^*G so that we have a Darboux coordinate system (X^i, λ_i). Keeping $\overrightarrow{\mathcal{H}_F}$ as general as possible, we can write it in terms of a Darboux basis as

$$\overrightarrow{\mathcal{H}_F} = \sum_{i=1}^{n} \left(A^i \frac{\partial}{\partial X^i} + B_i \frac{\partial}{\partial \lambda_i} \right) \tag{2.43}$$

for a series of coefficients A^i, B_i that have yet to be determined. By Definition 2.3.5, $\overrightarrow{\mathcal{H}_F}$ must satisfy $\iota_{\overrightarrow{\mathcal{H}_F}}\omega = -d\mathcal{H}_F$. Since $\mathcal{H}_F$ is a function on T^*G, $d\mathcal{H}_F$ is just a 1-form and can easily be computed in the Darboux basis as

$$d\mathcal{H}_F = \sum_{i=1}^{n} \left(\frac{\partial \mathcal{H}_f}{\partial X^i} dX^i + \frac{\partial \mathcal{H}_f}{\partial \lambda_i} d\lambda_i \right). \tag{2.44}$$

For notational simplicity, we will resort to the contraction notation for interior-products as they are more desirable than operating on a cotangent element. Thus take $\iota_X \omega = X \lrcorner \omega$ and calculate $\overrightarrow{\mathcal{H}_F} \lrcorner \omega$ to be [5]

$$\begin{aligned}
\overrightarrow{\mathcal{H}_F} \lrcorner \omega &= \sum_{j=1}^{n} \left(A^j \frac{\partial}{\partial X^j} + B_j \frac{\partial}{\partial \lambda_j} \right) \lrcorner \sum_{i=1}^{n} \left(dX^i \wedge d\lambda_i \right) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} \left[A^j \frac{\partial}{\partial X^j} \lrcorner \left(dX^i \wedge d\lambda_i \right) + B_j \frac{\partial}{\partial \lambda_j} \lrcorner \left(dX^i \wedge d\lambda_i \right) \right] \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} \left[A^j \left(dX^i \left(\frac{\partial}{\partial X^j} \right) d\lambda_i - d\lambda_i \left(\frac{\partial}{\partial X^j} \right) dX^i \right) + \right. \\
&\qquad \left. B_j \left(dX^i \left(\frac{\partial}{\partial \lambda_j} \right) d\lambda_i - d\lambda_i \left(\frac{\partial}{\partial \lambda_j} \right) dX^i \right) \right] \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} \left[A^j \delta^i_j d\lambda_i - B_j \delta^j_i dX^i \right] \qquad (2.45) \\
\overrightarrow{\mathcal{H}_F} \lrcorner \omega &= \sum_{i=1}^{n} \left(A^i d\lambda_i - B_i dX^i \right). \qquad (2.46)
\end{aligned}$$

In order to arrive at (2.45) we have used the orthonormality of the Darboux basis so that

$$d\lambda_i\left(\frac{\partial}{\partial\lambda_j}\right)=\delta_i^j$$
$$dX^i\left(\frac{\partial}{\partial X^j}\right)=\delta_j^i$$
$$dX^i\left(\frac{\partial}{\partial\lambda_j}\right)=0.$$

By equating (2.46) with the negative of (2.44) we see that $A^i=\dfrac{\partial\mathcal{H}_F}{\partial\lambda_i}, B_i=-\dfrac{\partial\mathcal{H}_F}{\partial X^i}$ so that locally, the Hamiltonian vector field has the form

$$\overrightarrow{\mathcal{H}_F}=\sum_{i=1}^{n}\left(\frac{\partial\mathcal{H}_F}{\partial\lambda_i}\frac{\partial}{\partial X^i}-\frac{\partial\mathcal{H}_F}{\partial X^i}\frac{\partial}{\partial\lambda_i}\right). \tag{2.47}$$

The new system given by $\dot{\xi}(t)=\overrightarrow{\mathcal{H}_F}(\xi(t))$ for $\xi(t)=(X,\lambda)\in T^*G$ will then have integral curves satisfying Hamilton's equations.

$$\frac{dX^i}{dt}=\frac{\partial\mathcal{H}_F}{\partial\lambda_i},\qquad \frac{d\lambda_i}{dt}=-\frac{\partial\mathcal{H}_F}{\partial X^i}. \tag{2.48}$$

Alternatively, if we specialize to the case where our manifold is a Lie-group we can further infer information about the structure of some of these elements. Let G be a Lie group with a Lie algebra $\mathfrak{g}$. We first recall that the pushforward of the right-translation action maps $dR_Y:T_XG\to T_{XY}G$, and hence the pullback acts as $(dR_Y)^*:T^*_{XY}G\to T^*_XG$. By a particular choice of $Y=X^{-1}$ then $(dR_{X^{-1}})^*:T^*_{\mathrm{id}}G\to T^*_XG$, and acts as

$$(dR_{X^{-1}})^*(\lambda)(H)=\lambda\left(dR_{X^{-1}}(H)\right),\qquad \forall\lambda\in T^*_{\mathrm{id}}G,\ H\in T_XG. \tag{2.49}$$

This implies that for every $\lambda\in T^*_{\mathrm{id}}G=\mathfrak{g}^*$ and $X\in G$ we have an element $(dR_{X^{-1}})(\lambda)\in T^*_XG$. Since X and λ were chosen arbitrarily, we can consider the identification $(X,\lambda)\leftrightarrow(dR_{X^{-1}})^*(\lambda)$ which allows us to write

$$T^*G\cong G\times\mathfrak{g}^*. \tag{2.50}$$

As we have seen, there are times when it is prudent to work with $T(T^*G)$. Since $T^*G\cong G\times\mathfrak{g}^*$ then

$$T(T^*G)=T(G\times\mathfrak{g}^*)=TG\oplus T\mathfrak{g}^*. \tag{2.51}$$

By applying the same analysis that resulted in (2.50) we can write $TG \cong G \times \mathfrak{g}$ and $T\mathfrak{g}^* \cong \mathfrak{g}^* \times \mathfrak{g}^*$ yielding

$$T(T^*G) \cong G \times \mathfrak{g} \times \mathfrak{g}^* \times \mathfrak{g}^*. \tag{2.52}$$

Just as we associate to a tangent bundle TG the pair (x, V) of a point $x \in G$ and the tangent vector V, we may choose to write this decomposition as $T(T^*G) = (G \times \mathfrak{g}^*) \times (\mathfrak{g} \times \mathfrak{g}^*)$ so that the base point is given by $(G \times \mathfrak{g}^*)$ and the "tangent vector" is given by $\mathfrak{g} \times \mathfrak{g}^*$, with a typical element denoted as $((X, \lambda), (H, J^*))$. When the base point is implied, we may just represent this as (H, J^*).

Given a function $f \in C^\infty(T^*G)$ and a vector field $V = (H, J^*) \in \Gamma(T(T^*G))$ it can be shown [39] that

$$(Vf)(X, \lambda) = \left(dR_X^* \frac{\partial f}{\partial X}(X, \lambda) \right) \left(H(X, \lambda) \right) + J^*(X, \lambda) \left(\frac{\partial f}{\partial \lambda}(X, \lambda) \right). \tag{2.53}$$

Consequently, if $\overrightarrow{\mathcal{H}_F} = (H(X, \lambda), J^*(X, \lambda)) \in \Gamma(T(T^*G))$ is the Hamiltonian vector field of the Hamiltonian $\mathcal{H}_F \in C^\infty(T^*G)$ we get that

$$H(X, \lambda) = \frac{\partial H}{\partial \lambda}(X, \lambda), \qquad J^*(X, \lambda) = -dR_X^* \left(\frac{\partial H}{\partial X}(X, \lambda) \right) - \operatorname{ad}^* H(\lambda) \tag{2.54}$$

which results in the generalized Hamilton's equations

$$\frac{dX}{dt} = dR_X \left(\frac{\partial H}{\partial \lambda} \right), \qquad \frac{d\lambda}{dt} = -dR_X^* \left(\frac{\partial H}{\partial X} \right) - \left(\operatorname{ad}^* \frac{\partial H}{\partial \lambda} \right) \lambda. \tag{2.55}$$

We are now in a position to examine the generalized Pontryagin principle. Let G be a smooth manifold and $\mathcal{D}$ be a distribution TG. Let f_0 be a cost criterion so that $(\mathcal{D}, f_0)$ defines a generalized optimal control problem. If we assume that $\mathcal{D}$ is given by a set of vector fields F_u then we have a system that can be represented by (2.31). We define a new variable $\tilde{X}$ which satisfies

$$\frac{d\tilde{X}}{dt} = f_0(X(t), U(X, t)) \tag{2.56}$$

and define an extended control system on $(X, \tilde{X}) \in M \times \mathbb{R}$ given by

$$\frac{d}{dt} X(t) = F(X(t), u(X(t), t)), \qquad \frac{d}{dt} \tilde{X}(t) = f_0(X(t), U(X(t), t)) \tag{2.57}$$

which is equivalent to $(\mathcal{D}, f_0)$. For a fixed control u, we can then extend the vector field to $\tilde{F}_u = (F(X, u), f_0(X, u)) \in T^*(\mathbb{R} \times G)$. Since $T^*\mathbb{R} \cong \mathbb{R}^2$ it follows that

$$T^*(\mathbb{R} \times G) \cong T^*\mathbb{R} \oplus T^*G \cong \mathbb{R}^2 \oplus T^*G. \tag{2.58}$$

Hence we can write $\xi \in T^*(\mathbb{R} \times G)$ as $\xi = \Big((X_0, \lambda_0), (X, \lambda)\Big)$ and the Hamiltonian of $\tilde{F}_u$ can then be written as

$$\mathcal{H}_{\lambda_0}(\xi, u) = \lambda_0 f_0(X(t), u) + \lambda(F(X(t), u)) \tag{2.59}$$

allowing us to state the following theorem:

Theorem 2.3.7 (The Maximum Principle, Jurdjevic [39])**.** *Suppose that $X(t)$ is an optimal trajectory of the generalized optimal control problem $(\mathcal{D}, f_0)$ defined on $[0, T]$. Let F be any control section of $\mathcal{D}$ that contains $X(t)$ and $u(t)$ be the control that generates $X(t)$. Then $\exists \lambda_0 \leq 0$ and an integral curve $\xi(t)$ of $\overrightarrow{\mathcal{H}_{\lambda_0}}(\cdot, u(t))$ where $\mathcal{H}_{\lambda_0}$ is given by (2.59), defined on all of $[0, T]$, such that*

1. *$X(t)$ is the projection of $\xi(t), \forall t \in [0, T]$; that is, if $\pi : T^*G \to G$ then $X(t) = \pi(\xi(t))$.*
2. *For any $t \in [0, T]$ such that $\lambda_0 = 0$ it follows that $\xi(t) \neq 0$.*
3. *$u(t)$ satisfies $\mathcal{H}_{\lambda_0}(\xi(t), u(t)) = \sup_{v \in \mathcal{U}} \mathcal{H}_{\lambda_0}(\xi(t), v)$*
4. *$\mathcal{H}_{\lambda_0}(\xi(t), u(t)) = 0, \forall t \in [0, T]$.*

We would like to point out the structural similarities between Theorem 2.3.7 and Theorem 2.1.5. This is largely due to the fact that most proofs of a manifold Maximum Principle follow from the proof of the original principle. A notable alternative to these traditional methods is given in [21] which uses a combination of the Whitney Embedding Theorem mentioned earlier, the Tubular Neighbourhood Theorem[52] and the PMP in $\mathbb{R}^n$.

The advantage of using the Maximum Principle as extended to smooth manifolds is that we have utilized the tools of differential geometry; a set of additional resources with which to analyze our problem. All the results given by the CMMP (Theorem 2.1.9) and the PMP still hold in the special case that our manifold is some subset of $\mathbb{R}^n$, but we can now avoid large dimensional embeddings into $\mathbb{R}$ and the resulting cumbersome computations by exploiting the inherent properties of the manifold.

Let us take an additional step and examine the same set of calculations given in section 2.1 that ultimately led to the definition of singular controls in Definition 2.1.7. Let G be a smooth manifold and assume that the function $F(X, u)$ is an m-input, control affine function that can be expressed as the right-translation of Lie-algebraic elements. Furthermore,

assume that the control amplitudes are symmetrically bounded $|u_i(t)| \leq M_i$. Consider a time-optimal cost function and assume that $H_0, H_i \in \text{Lie}(G), i = 1, \ldots, m$ so that

$$F(X, u) = (R_X)_* \left(H_0 + \sum_{i=1}^{m} u_i H_i \right) = \left(H_0 + \sum_{i=1}^{m} u_i H_i \right) X. \tag{2.60}$$

Notice that since $\text{Lie}(G)$ is an algebra it is closed under arbitrary sums ensuring that equation (2.60) is well-defined. Equation (2.31) can be written as

$$\frac{d}{dt} X(t) = \left(H_0 + \sum_{i=1}^{m} u_i(t) H_i \right) X(t). \tag{2.61}$$

By applying Theorem 2.3.7, we are guaranteed the existence of $\lambda \in T^*G$ which acts as a *linear* functional, giving the Pontryagin Hamiltonian (2.59)

$$\begin{aligned} \mathcal{H}_{\lambda_0}(\xi, u) &= \lambda_0 + \lambda \left[\left(H_0 + \sum_{i=1}^{m} u_i(t) H_i \right) X(t) \right] \\ &= \lambda_0 + \lambda(H_0 X(t)) + \sum_{i=1}^{m} u_i(t) \lambda(H_i X(t)) \\ &= \lambda_0 + \psi_0(t) + \sum_{i=1}^{m} u_i(t) \psi_i(t). \end{aligned}$$

By applying condition 3 of Theorem 2.3.7 we see that our control is still governed by $u_i(t) = \text{sgn}(\psi_i(t)) M_i$ whenever sgn is well-defined. Precisely the same definition of singular, normal, and abnormal controls from Definition 2.1.7 and 2.1.8 still applies.

2.4 Casting Quantum Mechanics into a Control Framework

With the control theoretical part established, we can now move to casting quantum mechanics into a formalism that permits control techniques to be used. We will begin by introducing some of the more fundamental concepts of quantum theory and how the geometric framework can be built around these ideas. Afterwards, we will discuss the control domain wherein the problem will be formalized and some results stated.

2.4.1 Introduction to Quantum Mechanics

A large part of the power of quantum mechanics lies in the rigorous mathematical framework which has been established around the physical observations. As a matter of fact, once one axiomatically defines a few postulates of closed quantum systems, the rest of the theory can be built. While the abstract formulation of quantum mechanics into a purely mathematical framework might remove the physical implications of what is occurring, it allows us to ignore the physical details and embed our problem purely in terms of a differential geometric setting. We present below an abstract introduction to some of the most fundamental results pertaining to quantum systems which will be relevant to our discussion.

Definition 2.4.1. Let $\mathcal{Z}$ be a complex vector space and $\langle \cdot, \cdot \rangle : \mathcal{Z} \times \mathcal{Z} \to \mathbb{C}$ a positive-definite sesquilinear form. We shall refer to $d(x, y) = \sqrt{\langle x - y, x - y \rangle}$ as the *induced metric*, which satisfies all of the properties of a metric under either formalism. The pair $(\mathcal{Z}, \langle \cdot, \cdot \rangle)$ is said to be a *Hilbert space* if $(\mathcal{Z}, d)$ is complete in d.

It is worth noting there is a degree of personal preference in choosing whether the first or second argument is conjugate linear. In quantum mechanics, one often takes the second argument to be linear and hence this is the definition we will use throughout the remainder of this thesis.

While Hilbert spaces are used extensively in quantum mechanics, there is a subtle difference in how the inner-product is used. In order to properly discuss why this is the case, it is necessary to introduce the following theorem:

Theorem 2.4.2 (Riesz Theorem, Kreyszig [49])**.** *Let $(\mathcal{Z}, \langle \cdot, \cdot \rangle)$ be an inner-product space and consider the dual space $\mathcal{Z}^*$ of linear functionals on $\mathcal{Z}$. Every bounded linear functional f can be represented in terms of an inner-product; namely, there exists a unique $z \in \mathcal{Z}$ such that $f(x) = \langle x, z \rangle$ and $\|z\| = \|f\|$.*

We note that the mapping $\Phi : \mathcal{Z}^* \to \mathcal{Z}$ sending $\Phi f = z$ is an anti-linear isometric isomorphism between the Hilbert space and its dual . This allows us to move between an inner-product and functional evaluation on Hilbert spaces.

The Dirac *bra-ket* notation is standard throughout quantum mechanics and so it is introduced here. Given a state $\psi \in \mathcal{Z}$ we denote this by the *ket* $|\psi\rangle$. Alternatively, given $\phi \in \mathcal{Z}^*$ we denote this by the *bra* $\langle \phi |$. It is here that Riesz's theorem becomes valuable. Let $\varphi = \Phi\phi$, and notice that

$$\langle \phi | \psi \rangle = \langle \varphi, \psi \rangle \tag{2.62}$$

where the left-hand-side is the evaluation of a linear functional on an element of a Hilbert space, and the right-hand-side is an inner-product. In an abuse of notation, we associate the image of a bra under Φ to its corresponding ket. More precisely, we define

$$\Phi \langle \phi | = | \phi \rangle$$

so that $\langle \phi | \psi \rangle = \langle \phi, \psi \rangle$. Note that if U is an operator on $\mathcal{H}$, then the notation $\langle \phi \, | U | \, \psi \rangle$ is taken to mean $\langle \phi, U\psi \rangle$.

The unitary group is an important algebraic structure in quantum mechanics, and will be very crucial to this thesis.

Definition 2.4.3. Let $(\mathcal{Z}, \langle \cdot, \cdot \rangle)$ be a complex Hilbert space. A *unitary transformation* on $\mathcal{Z}$ is a surjective linear isometry. More precisely, if $u, v \in \mathcal{Z}$ then $U : \mathcal{Z} \to \mathcal{Z}$ is unitary if

$$\langle u, v \rangle = \langle Uu, Uv \rangle . \tag{2.63}$$

In the event that $\mathcal{Z} = \mathbb{C}^n$ for some $n \in \mathbb{N}$, we define the *unitary group* as

$$\mathfrak{U}(n) = \left\{ U \in M_n(\mathbb{C}) \;\middle|\; U^\dagger U = U U^\dagger = id_n \right\} . \tag{2.64}$$

We define the *special unitary group* $\mathfrak{SU}(n)$ as the subset of $\mathfrak{U}(n)$ with unit determinant

$$\mathfrak{SU}(n) = \left\{ V \in \mathfrak{U}(n) \;\middle|\; \det V = 1 \right\} . \tag{2.65}$$

For Lie algebraic reasons that will be discussed in the future, we define the space of skew-Hermitian matrices

$$\mathfrak{u}(n) = \left\{ U \in M_n(\mathbb{C}) \;\middle|\; U + U^\dagger = 0 \right\} \tag{2.66}$$

and the subset of traceless skew-Hermitian matrices

$$\mathfrak{su}(n) = \left\{ U \in \mathfrak{u}(n) \;\middle|\; \mathrm{Tr}[U] = 0 \right\} . \tag{2.67}$$

Definition 2.4.4. Let $(\mathcal{Z}, \langle \cdot, \cdot \rangle)$ be a complex Hilbert space. A *phase factor* is an element of the circle group $\mathbb{T} = \mathfrak{U}(1)$ in $\mathbb{C}$.

The group $\mathfrak{U}(1)$ corresponds to the circle group, or equivalently the set of all complex numbers with modulus 1. We can represent elements of $\mathfrak{U}(1)$ as $e^{i\theta}$ for some $\theta \in [0, 2\pi)$, which informs us that the induced norm $\|\psi\| = \sqrt{\langle \psi | \psi \rangle}$ is invariant under global phase since

$$\left\| e^{i\theta} \psi \right\|^2 = \left\langle e^{i\theta} \psi | e^{i\theta} \psi \right\rangle = e^{i\theta} e^{-i\theta} \langle \psi | \psi \rangle = \langle \psi | \psi \rangle = \|\psi\|^2 . \tag{2.68}$$

In the mathematical formulation of the theory of quantum mechanics, one axiomatically assumes that every closed quantum system can be described as existing in an ambient, separable, complex Hilbert space. Furthermore, we have seen that the concept of *global* phases play no part in the mathematical formalism (though we note that *relative phases* are important and observable). For this reason, we can instead consider the equivalence class of elements modulo a phase-factor which is equivalent to working with the the projective Hilbert space $\mathcal{P}(\mathcal{Z})$.

As the primary focus of this treatise will be to examine applications to quantum computing, we will restrict ourselves to the case where $\mathcal{Z}$ is finite dimensional. It is shown in Appendix C.1 that two Hilbert spaces are isomorphic if and only if they have the same Hilbert dimension. As a consequence, one may assume for the remainder of this thesis that $\mathcal{Z} = \mathbb{C}^n$ and $\mathcal{P}(\mathcal{Z}) = \mathbb{PC}^{n-1}$ unless stated otherwise. We note the particular instance when $n = 2$ in which case $\mathbb{PC}^1$ is the *Bloch Sphere*.

Definition 2.4.5. Let $(\mathcal{Z}, \langle\cdot|\cdot\rangle)$ be a Hilbert space and $|\psi_0\rangle \in \mathcal{Z}$. Let $H(t)$ be an operator that describes the quantization of the system energy. The *Schrödinger equation* is a dynamical equation that governs the evolution of the state $|\psi_0\rangle$ in the closed system described by both $\mathcal{Z}$ and $H(t)$ and is given by

$$i\hbar\frac{d}{dt}|\psi(t)\rangle = H(t)\,|\psi(t)\rangle\,, \qquad |\psi(t_0)\rangle = |\psi_0\rangle \tag{2.69}$$

where $\hbar$ is the reduced Planck's constant and $i^2 = -1$.

In the case where $H = H(t)$ is time-independent, this can be solved using an exponential function. Unfortunately, it is often not suitable to assume a time-independent Hamiltonian (as will be the case in control theory) which necessitates the introduction of the *time-ordering* operator $\mathcal{T}$ which allows us to keep the exponential notation. By applying the theory of differential equations, we can write the general solution to (2.69) as

$$|\psi(t)\rangle = \mathcal{T}\exp\left[-\frac{i}{\hbar}\int_{t_0}^{t} H(t)\;dt\right]|\psi_0\rangle\,. \tag{2.70}$$

Defining $X(t,t_0) = \mathcal{T}\exp\left[-\frac{i}{\hbar}\int_{t_0}^{t} H\;dt\right]$ we get that we can propagate any initial state $|\psi_0\rangle$ to its final state simply by left-multiplication $|\psi_0\rangle \mapsto X(t,t_0)\,|\psi_0\rangle$. For this reason, we often refer to $X(t,t_0)$ as the time-propagation operator from t_0 to t. We notice that the choice of $|\psi_0\rangle$ was arbitrary and that the particular form of $X(t,t_0)$ depends only on the choice of Hamiltonian $H(t)$. Consequently, instead of considering a different problem for

each choice of initial condition $|\psi_0\rangle$, it is more prudent to instead examine the operator $X(t, t_0)$ itself.

To deduce the differential equation for $X(t, t_0)$ it is a simple matter to substitute $|\psi(t)\rangle = X(t, t_0)\,|\psi_0\rangle$ into (2.69) to get

$$i\hbar \frac{d}{dt} X(t, t_0)\,|\psi_0\rangle = H(t) X(t, t_0)\,|\psi_0\rangle\,. \tag{2.71}$$

Since $|\psi_0\rangle$ is constant in time, we can remove it from the differential equation. Furthermore, in order to ensure that $|\psi(t_0)\rangle = |\psi_0\rangle$ we demand that $X(t_0, t_0) = \mathrm{id}$, the identity operator. Summarizing, we see that equation (2.69) is equivalent to

$$i\hbar \frac{d}{dt} X(t, t_0) = H(t) X(t, t_0), \qquad X(t_0, t_0) = \mathrm{id}. \tag{2.72}$$

We often assume that t_0 is fixed and often even 0. This allows us to remove the dependency of X on t_0 instead writing $X(t) = X(t, t_0)$.

Our first thought might be to consider the control of pure quantum states subject to a specific Hamiltonian. If one chooses to be even more general, permitting the existence of mixed states, then we might consider evolution in the space of density matrices. However, in the closed-quantum formalism, we benefit from two interpretations of time-evolution: the Schrödinger and Heisenberg pictures. The Schrödinger picture views states as dynamic variables with static operators, whilst the Heisenberg picture allows operators to progress through time instead. By utilizing the Heisenberg picture, we do not need to worry about whether we are dealing with pure or mixed states, since by explicitly giving an expression for $X(t, t_0)$ we can characterize both instances.

Physically, we require that energy levels of quantum systems be given by real numbers. Since the measurement of such levels corresponds to eigenvalues of the matrix representation, this puts a restriction on the system Hamiltonian H.

We note that the quantum Hamiltonian is necessarily Hermitian. The expectation value of a measurement of energy given that our ensemble is prepared in the state $|\psi\rangle$ is given by $\langle\psi\,|H|\,\psi\rangle$. Physically, we demand that such measurements yield real results. This means that

$$\langle\psi\,|H|\,\psi\rangle = \langle\psi\,|H|\,\psi\rangle^*\,. \tag{2.73}$$

However, by conjugate symmetry, $\langle\psi\,|H|\,\psi\rangle^* = \langle H\psi|\psi\rangle$ giving

$$\langle H\psi|\psi\rangle = \langle\psi|H\psi\rangle\,. \tag{2.74}$$

It follows that $H = H^\dagger$ and so H is Hermitian.

We denote the set of $N \times N$ Hermitian matrices as $i\mathfrak{u}(N)$. Lie algebraic methods then allow us to conclude that the trajectory $X(t)$ must be unitary for all time, since the Schrödinger equation implies that iH is in the Lie-algebra for X, so $X \in \mathfrak{U}(N)$.

The rest of this thesis will focus on the Lie groups $\mathfrak{U}(N)$ or its subgroup $\mathfrak{SU}(N)$. It is important to note that these Lie groups are also complex Hilbert spaces under the Hilbert-Schmidt norm given by

$$\langle X, Y \rangle = \mathrm{Tr}\left[X^\dagger Y\right] \tag{2.75}$$

which is derived by embedding $(\mathfrak{S})\mathfrak{U}(N)$ as a $\mathbb{C}$-vector space under the standard Euclidean inner-product. As mentioned before, this inner product is proportional to the Killing form on $(\mathfrak{S})\mathfrak{U}(N)$.

Proposition 2.4.6. *If we are given the inner-product*

$$\langle X, Y \rangle = \mathrm{Tr}\left[X^\dagger Y\right]$$

then $(\mathfrak{U}(N), \langle \cdot, \cdot \rangle)$ *is a Hilbert space.*

Proof. It is well known that the unitary group $\mathfrak{U}(N)$ is compact when viewed as a subspace of $\mathbb{R}^{N^2}$ under the standard Euclidean inner-product. Since (2.75) is equivalent to the Euclidean inner-product, it follows that $\mathfrak{U}(N)$ is compact under the induced metric of $\langle X, Y \rangle$. It is then complete since all compact spaces are complete. □

Corollary 2.4.7. *The special unitary group* $\mathfrak{SU}(N)$ *is also a Hilbert space under* $\langle X, Y \rangle$.

Proof. It suffices to show that the special unitary group is a closed subset of $\mathfrak{U}(N)$, since closed subsets of compact sets are compact. It is trivially a subset by definition, so it remains to show that it is closed. By [8], we know that the determinant mapping

$$\det : M_N(\mathbb{C}) \to \mathbb{R} \tag{2.76}$$

is a continuous mapping. We can define $\mathfrak{SU}(N) = \det^{-1}\{1\} \cap \mathfrak{U}(N)$. Since det is continuous, $\det^{-1}\{1\}$ is closed. Since $\mathfrak{U}(N)$ is compact it is also closed. Thus we have that $\mathfrak{SU}(N)$ is the intersection of two closed sets and hence is closed. □

Proposition 2.4.8. *With the inner-product* $\langle X, Y \rangle$ *as defined in* (2.75)*, the induced metric is given by*

$$d(X, Y) = \sqrt{2N - 2\Re \langle X, Y \rangle}. \tag{2.77}$$

Proof. It is well known that all inner-products induce norms and all norms induced metrics. By transitivity, the squared-metric is then given by

$$d(X,Y)^2 = \|X-Y\|^2 = \langle X-Y, X-Y\rangle\,. \tag{2.78}$$

Expanding the right-hand side, we get

$$\begin{aligned}\langle X-Y, X-Y\rangle &= \langle X,X\rangle + \langle Y,Y\rangle - \langle X,Y\rangle - \langle Y,X\rangle \\ &= \|X\|^2 + \|Y\|^2 - \langle X,Y\rangle - \overline{\langle X,Y\rangle} \\ &= \|X\|^2 + \|Y\|^2 - 2\Re\langle X,Y\rangle\,. \end{aligned}\tag{2.79}$$

By definition of the norm, for any $X \in \mathfrak{U}(N)$

$$\|X\|^2 = \langle X,X\rangle = \mathrm{Tr}[X^\dagger X] = \mathrm{Tr}[\mathrm{id}] = N \tag{2.80}$$

and so we conclude that

$$d(X,Y)^2 = 2N - 2\Re\langle X,Y\rangle \tag{2.81}$$

which gives the desired result. □

It is important to note here that the inner-product defined by the Killing form is natural when viewing $\mathfrak{U}(N)$ as a subspace of $\mathbb{R}$. However, this is not the only way of measuring distances between unitary operators. Another commonly used function is the *diamond norm*. In fact, there are many different metrics which can be used, and so our choice of using a trace inner-product is not unique.

2.4.2 Quantum Control

Since the Schrödinger equation describes the evolution of closed quantum systems, we focus our attention on describing its properties and enforcing our control through its dynamics. The first thing that we note is that (2.72) is right-invariant.

Proposition 2.4.9. *The Schrödinger equation is right-invariant on* $\mathfrak{U}(n)$.

Proof. Let $X(t)$ be a solution to (2.72) and choose an arbitrary constant matrix $M \in M_n(\mathbb{C})$. By defining $Y(t) = X(t)M$ it follows that

$$i\hbar\frac{d}{dt}Y(t) = i\hbar\frac{dX}{dt}(t)M = HX(t)M = HY(t) \tag{2.82}$$

which is precisely what is required. □

As discussed in Section 2.2.5, controllability is now easy to verify as controllability from the identity characterizes controllability over the entire space.

We further wish to be able to control the system in a *coherent* fashion; that is, by modifying properties of the system rather than the state itself. This can be done by manipulating the system Hamiltonian. We can write it in terms of its internal and external contributions

$$H = H_{\text{int}} + H_{\text{ext}} \tag{2.83}$$

whose explicit expression often takes the form of an m-input, affine control system

$$H = H_0 + \sum_{j=1}^{m} u_j(t) H_j \tag{2.84}$$

where $H_i \in \mathfrak{U}(N)$ for each $i = 0, \ldots, m$. Physically, $H_0 = H_{\text{int}}$ describes the *drift* Hamiltonian that governs the system dynamics in the absence of control, $u_j(t) \in \mathbb{R}$ describes the control amplitudes, and H_j are the control Hamiltonians describing how we are physically exerting our presence the system. Note that in the event that $H_0 = 0$ we say that the system is *driftless*.

Demanding that the system obey an affine structure is a natural consequence arising from the additive nature of energy functions. In particular, introducing an additional source of energy to a system results in an additive change in the system Hamiltonian. Since we have chosen to exercise our control by manipulating external energy sources, the affine restriction further allows us to benefit from the extensive work previously done in affine control theory. Additionally, it is important to note that we have taken the H_j to be time-independent, which we are permitted to do from a design perspective and corresponds to keeping the direction of the fields constant.

In the following, we will quickly review some of the work done by [37] in reviewing the specialization of affine control systems to the unitary group.

Consider the differential system given by (2.72) and write $H = H(u)$ to denote the explicit dependence of the Hamiltonian on the control variables. Furthermore, let us specialize to a time-optimal problem and choose units such that $\hbar = 1$. By Theorem 2.3.7 it follows that $\frac{d}{dt}\lambda(t) = -\frac{\partial \mathcal{H}_H}{\partial X}$ where

$$\mathcal{H}_H((X, \lambda), u) = \lambda_0 + \Re\lambda(t)(-iH(u)X(t)). \tag{2.85}$$

In [5, 37] the authors choose to represent the functional action of λ as in inner-product, writing $\lambda(X) = \langle \lambda, X \rangle$. We note that even when this inner-product is not explicit, this

notation is still acceptable as it represents a bilinear mapping that preserves the duality shared by λ and X. This allows us to write

$$\mathcal{H}_H((X,\lambda),u) = \lambda_0 + \Re\left\langle \lambda(t), -iH(u)X(t)\right\rangle. \tag{2.86}$$

By utilizing Hamilton's equations given in (2.48), one can show [37] that $\lambda(t)$ satisfies $\dot{\lambda}(t) = -iH(t)\lambda(t)$ with solution given by $\lambda(t) = X(t)\lambda(0)$. By making use of the adjoint of $X(t)$, we can write (2.86) as

$$\mathcal{H}_H((X,\lambda),u) = \lambda_0 + \Re\left\langle \lambda(0), -iX^\dagger(t)H(u)X(t)\right\rangle \tag{2.87}$$

from which the author claims that $\lambda(0)$ can be replaced by an element $M \in \mathfrak{su}(N)$ giving

$$\mathcal{H}_H((X,\lambda),u) = \lambda_0 + \Re\left\langle M, -iX^\dagger(t)H(u)X(t)\right\rangle. \tag{2.88}$$

We note that when the inner-product is taken to be the trace inner-product, we can drop the real-argument from (2.88) since the trace of the product of two elements in $\mathfrak{u}(n)$ is guaranteed to be real.

The ability to convert the Hamiltonian into the form given by (2.88) allowed the author of [37] to consider a partial treatment of optimal extremals. However, the general argument used to claim the existence of the matrix M was purely existential and it is not yet known how to construct it explicitly. Without further information regarding the structure of M, we are unable to exploit this simplification to help in our computational or analytical analysis.

Chapter 3

Literature Review

While the focus of this thesis pertains to time-optimal, bounded control of unitary operators, we will survey some of the prominent results in closely related fields. Such results will be important to our study as they operate through fundamental assumptions, provide similar techniques, or pose potential for future development. This treatment will be followed by a more detailed presentation of literature relevant to the thesis, before moving to an introduction of the mathematical theory we will be using.

3.1 General Optimal Quantum Control

The purpose of optimal control is to minimize a cost function while driving an initial system configuration to a terminal configuration. Optimal *quantum control* is an incipient field whose results stem from only a few decades of serious research. The burgeoning of technological advances that have allowed experimentalists to manipulate the intricacies of microscopic systems has flourished activity in quantum control research. The breadth of the field precludes a comprehensive examination despite its nascence, though we hope to explore some interesting results that pertain, although indirectly, to this thesis.

This section contains a summary of the literature in fields that are tangentially related to the thesis topic. Much of the mathematical details will be omitted in order to keep the presentation cogent and to limit the amount of unnecessary background information required. The reader interested in the tedium is referred to the papers themselves, as well as the appendix whenever more complex mathematics is unavoidable.

3.1.1 Controllability

We shall start by discussing the issue of controllability; that is, whether any state is accessible from any other state. This topic has received a great deal of consideration because of how fundamentally important it is to the study of any control system, though a plethora of results has made the problem manageable. Controllability determines our ability to access particular states within the system. In ideal situations, a system is completely controllable which tells us that any state can be driven to any other.

As it pertains to our study, we refer to the seminal paper by Jurdejevic and Sussman [40] who aimed to establish controllability results on general Lie groups. This paper is very important to studying controllability on Lie groups and established the foundations of such a study.

The authors consider multiple sets of admissible controls for a right-invariant control-affine system with drift. To be more explicit, let G be a Lie group and $X(t) \in G$ describe a time varying state. Assume further that the state obeys the system dynamics given by

$$\frac{d}{dt}X(t) = H_0(X(t)) + \sum_{i=1}^{m} u_i(t)H_i(X(t)) \tag{3.1}$$

where $\{H_i\}_{i=0}^m$ is a set of prescribed right-invariant vector fields. Theorem 2.2.28 tell us that the Lie algebra $\mathfrak{g}$ can be identified with the tangent space at identity, so we can write $\mathfrak{g} \cong T_eG$ and $\{H_i\}_{i=1}^m \subseteq T_eG$.

We say that $\{H_i\}_{i=1}^m$ *generates* the Lie algebra if the distribution spanned by $\{H_i\}_{i=1}^m$ is bracket generating. More precisely, if $\mathcal{D} = \operatorname{span}\{H_i\}_{i=1}^m$ then

$$\mathcal{D} \subseteq \mathcal{D}^2 \subseteq \ldots \subseteq \mathcal{D}^r = TG$$

for some $r \in \mathbb{N}$. Jurdjevic and Sussman show that if G is connected and $\{H_i\}_{i=1}^m$ is bracket generating then (3.1) is controllable. Furthermore, if G is compact or driftless the converse also holds.

The paper also gives a structure argument in that $\forall X \in G$ we can write X as a finite product of elements of the form $\exp[\tau H_i(X(t))]$, and that whenever G is connected and (3.1) is reduced to a driftless, two-input system

$$\frac{d}{dt}X(t) = u_1(t)H_1(X(t)) + u_2(t)H_2(X(t)), \tag{3.2}$$

almost any pair of control fields will generate the entire Lie algebra. This means that almost any two right-invariant vector fields give rise to a controllable system. Mathematically, the precise statement is that

$$\left\{(H_1, H_2) \in \mathfrak{g} \times \mathfrak{g} \;\middle|\; H_1, H_2 \text{ generates } \mathfrak{g}\right\} \tag{3.3}$$

is open and dense in $\mathfrak{g} \times \mathfrak{g}$.

The previous paper truly encapsulates the pertinent results required for application to control of operators on closed quantum systems. Such operators evolve on the unitary or special unitary groups, which are themselves connected and compact. The property of the control fields be bracket generating is known as the *Lie Algebra Rank Condition* (LARC) and remains the primary method for testing controllability.

It should be noted that despite being revered for its simplicity, the LARC can be a computationally expensive proposition. From Definition 2.2.40, determining whether a distribution is bracket generating requires computing many Lie brackets. The number of computations that need to be calculated can be reduced by realizing that if $\mathcal{F}$ and $\mathcal{G}$ are two distributions with bases $\{f_i\}_{i=1}^{m_1}, \{g_i\}_{i=1}^{m_2}$ then

$$[\mathcal{F}, \mathcal{G}] = \operatorname{span}\left\{[f, g] \;\middle|\; f \in \mathcal{F}, g \in \mathcal{G}\right\} = \operatorname{span}\left\{[f_j, g_k] \;\middle|\; j = 1, \ldots, m_1, k = 1, \ldots, m_2\right\}.$$

One needs to iteratively apply this computation to $\mathcal{D}$ while maintaining knowledge of the basis vectors[67]. Without an *a priori* knowledge of whether the set is bracket generating, it is uncertain whether this algorithm will even terminate. This is exacerbated by the fact that $\dim_{\mathbb{R}} \mathfrak{U}(d) = 4^d$, and so as we consider multi-particle systems, the number of basis elements needed to span TG grows exponentially. For this reason, Altafini[6] has worked on alternate methods to test controllability without computing Lie brackets.

Altafini's method relies heavily on the root-space decomposition of the manifold and properties of the graphical representation of the control field, a tool which we will also see implemented in the time-optimal considerations of [4]. Since Altafini's primary interest is the application to quantum control, he considers the compact Lie group $\mathfrak{SU}(N)$ and the single-input affine control system

$$\frac{d}{dt}X(t) = (H_0 + u(t)H_1)X(t) \tag{3.4}$$

where $H_0, H_1 \in \mathfrak{su}(N)$, the real Lie algebra of traceless skew-Hermitian operators. Let $\mathfrak{su}(N)^{\mathbb{C}}$ denote the complexification of $\mathfrak{su}(N)$, and take $h \in \mathfrak{su}(N)^{\mathbb{C}}$. We say that h is

regular if $\dim(\ker \mathrm{ad}_H) = N-1$ where $\mathrm{ad}_H = [H, \cdot]$ is the adjoint representation of H. The element h is *strongly regular* if all non-zero eigenvalues of ad_H have unit multiplicity. For the interested reader, we note that eigenvalues of the linear operator ad_H are also known as *weights* of the representation.

Fix an $h \in \mathfrak{su}(N)^{\mathbb{C}}$. The *Cartan subalgebra* corresponding to h is then the zero eigenspace of ad_H given by

$$\mathfrak{g}_0(h) = \left\{ k \in \mathfrak{su}(N)^{\mathbb{C}} \;\middle|\; \mathrm{ad}_h\, k = 0 \right\}. \tag{3.5}$$

The *roots* of $\mathfrak{su}(N)^{\mathbb{C}}$ under h are then the set of functionals $\alpha : \mathfrak{g}_0(h) \to \mathbb{C}$ such that $\mathrm{ad}_h\, k = \alpha(h)k$; that is, they are the functional eigenvalues. Hence we can decompose $\mathfrak{su}(N)^{\mathbb{C}}$ as

$$\mathfrak{su}(N)^{\mathbb{C}} = \mathfrak{g}_0(h) + \bigoplus_{\alpha \in \Delta} \mathfrak{g}_\alpha, \qquad \mathfrak{g}_\alpha = \left\{ k \in \mathfrak{su}(N)^{\mathbb{C}} \;\middle|\; \mathrm{ad}_h\, k = \alpha(h)k \right\} \tag{3.6}$$

where Δ denotes the set of all roots under h. Altafini claims that the roots have a physical interpretation, and correspond to transitions between different energy levels of the system.

Now assume that C is a square matrix and associate to it the graph $\mathcal{G}_C = (\mathcal{N}_C, \mathcal{C}_C)$ consisting of nodes $\mathcal{N}_C$ and oriented paths $\mathcal{C}_C$ determined by using C as an adjacency matrix. The graph $\mathcal{G}_B$ is said to be strongly connected if for every two nodes there is an oriented path connecting them.

The primary results of Altafini's paper are as follows: using this root-space decomposition, if H_0, H_1 are as in (3.4) with $\mathcal{G}_B$ strongly connected and H_0 strongly regular then the system is controllable. There are further developments in the paper that allow these conditions to be relaxed to a notion of relative regularity called H_1-strong regularity, as well a treatment of a few special cases that arise; however, these arguments are beyond the scope of this thesis. Ultimately, the author presents a different characterization for controllability on quantum mechanical systems that completely avoids the computation of Lie brackets. The theoretical description given by Altafini is more complicated than the LARC, but makes use of analytical techniques to give a result that is easier to verify numerically.

3.1.2 Control Landscapes and Geometry

Next we would like to consider some of the work done on control landscapes. The objective here is to discover the general topological properties of quantum control systems irrespective of any particular form of the system Hamiltonian.

A collection of two papers written by Nielsen et al. [58, 59] aim to relate the notion of quantum circuit complexity to geometric objects on Riemannian manifolds. The authors show that one can obtain bounds on circuit complexity by considering trajectories derived from related problems in optimal control. In particular, by using a time-optimal control problem, one can show that the implementation time and gate complexity are related by a polynomial which scales in the number of qubits.

The first of these two papers [59] does not explicitly consider the relationship between control theory and gate complexity, but instead lays the foundation that will be used in the second paper. The motivation is to impose a cost function on the Hamiltonian that, when considered as a decomposition into the Pauli basis, penalizes 3-or-more qubit operators. From this a Riemannian metric is constructed for which the gate complexity is polynomial in the induced length metric distance between the gate and identity. More precisely, if we have an n-qubit system, let

$$d(I, X) = \inf_{\gamma \in \Gamma_{I,X}} \int_0^1 \|\gamma'(t)\| \ dt, \qquad \Gamma_{I,X} = \{\gamma : [0,1] \to X, \gamma(0) = I, \gamma(1) = X\} \tag{3.7}$$

be the induced length metric between the identity I and another other $X \in \mathfrak{SU}(2^n)$. The authors show that it requires only $O(n^6 d(I,X)^3)$ one- and two-body gates to synthesize an operator X_c such that $\|X - X_c\| \leq c$ for any constant $c > 0$, where $\|\cdot\|$ is the operator norm.

The second paper [58] exploits the lack of generality of the first. In particular, while a very specific metric was used in the first paper, it was found that many other metrics could also be used. Consider a two level, n-qubit system occupying $\mathfrak{SU}(2^n)$. Let $X \in \mathfrak{SU}(2^n)$ and denote by $\mathcal{G}(X)$ the number of one- and two-qubit gates required to exactly synthesize X. Let $\mathcal{G}(X, \epsilon)$ denote the number of one- and two-qubit gates required to approximately synthesize X up to an error ϵ. Next, consider the operator Schrödinger equation

$$\frac{dU}{dt} = -iH(t)X(t); \qquad H(t) = \sum_{j=1}^{m} u_j(t) H_j \tag{3.8}$$

for control fields $\vec{u}(t) = (u_1(t), \ldots, u_m(t))$ and $H_j \in \mathfrak{su}(2^n)$, assumed to be controllable. Define the cost function $c : \mathbb{R}^m \to \mathbb{R}$ parameterized in terms of the control fields given in (3.8) over some admissible set $\Omega \in \mathbb{R}^m$. We then say that the *cost of the unitary* X_d is given by

$$C(X_d) = \inf_{T > 0, \vec{u} \in \Omega_{X_d, T}} \int_0^T c(\vec{u}(t)) \ dt \tag{3.9}$$

where $\Omega_{X_d,T} = \left\{ \vec{u} \in \Omega \;\middle|\; X(T) = X_d \right\}$ is the set of control fields that drive the system (3.8) to X_d in time T.

The results of the paper are very general and can be applied to many metrics. We focus our attention on the case of time-optimal control where $c(\vec{u}(t)) \equiv 1$, and use this to define a relationship between $C(X)$ and $\mathcal{G}(X)$. Consider again the single-input, affine control system with drift given by (3.4) with $|u(t)| \leq 1$ and $\{H_0, H_1\}$ chosen such that the system is controllable. Note that $C(X)$ now represents the optimal time to implement X with bounded controls. The authors show that there exists a polynomial $q(n)$ such that $C(X) \leq q(n)\mathcal{G}(X)$; that is, $C(X)$ and $\mathcal{G}(X)$ differ by at most a multiplicative factor that scales polynomially in n.

The authors use their general results to find other relationships using a sub-Riemannian metric and the same Riemannian metric used in [59]. While no other examples of control systems are given, the theorems provided in the paper provides a framework in which other bounds could be found by applying different cost functions. These same authors have continued their work in this area, though the connection to control theory once again becomes vague. The interested reader is referred to [26] for further reading.

There is another collection of works published annually by a series of authors concerning optimal control transition landscapes [33, 34, 35, 62, 63, 64, 74]. These papers consider a general landscape topology without referring to the form of the Hamiltonian. This is done by parameterizing the cost function purely in terms of the control fields, known as the kinematic approach. In particular, the goal of the first two papers is to analyze the possibility of sub-optimal local extrema when maximizing state transition probabilities [64] or gate synthesis [63]. This result is of particular importance to the study of the numerics of high-fidelity controls, and was introduced as an attempt to explain why experimental and numerical results often found good candidate solutions while avoiding local traps. We would like to emphasize here that the results found below do not apply directly to time-optimal control, since the objective is simply to synthesize the final state regardless of the time taken for implementation.

Consider an N-level system and let X_d be the desired unitary with X giving the best approximation to X_d. Since the exponential function is surjective, there must exist matrices A and B such that $X_d = \exp[iB]$ and $X = \exp[iA]$. Let $\{b_i, |\beta_i\rangle\}, \{a_i, |\alpha_i\rangle\}$ denote the eigen-sets of B and A respectively. We want to optimize the transition from X to X_d which is done by considering the cost

$$C(X) = 2N - 2\Re\mathrm{Tr}[X_d^\dagger X] \tag{3.10}$$

which is derived from the Frobenius norm in Section 2.4.1. However, the evolution of

X depends on the state of the Hamiltonian. By taking $u(t)$ as the control vector and reparameterizing X purely in terms of $u(t)$, we can write the cost function in terms of the controls as $C[u(t)]$, allowing us to examine how changes in $u(t)$ affect C without any explicit knowledge of the Hamiltonian.

The authors are able to show that the infinitesimal generators A and B commute so that they share a set of eigenvalues. Furthermore, the difference between any pair of ordered eigenvalues is an integer multiple of π. This means that for a fixed set of eigenvalues a_r and $b_r, \exists n_r \in \mathbb{N}$ such that $a_r - b_r = n_r\pi$. By expanding the trace operator in (3.10) in terms of the eigenbasis of B the authors are able to shown that

$$C = 2N - 2\sum_r (-1)^{n_r}. \tag{3.11}$$

Hence, depending on the overall parity of the n_r, it follows that C can only attain a discrete set of values corresponding to $C = 0, \ldots, 4N$. The case where $C = 0$ corresponds to a perfect but out-of-phase control $X_d = -X$ and $C = 4N$ is perfect in-phase control $X_d = X$.

The next step is to analyze the Hessian of C with respect to $u(t)$ to see if any results can be found concerning the existence of suboptimal extrema. It is shown that the Hessian is positive definite when $C = 4N$, negative definite when $C = 0$ and indefinite for all other values of C. This implies that all sub-optimal states correspond to "saddle points" and should not impede the ability of local search algorithms.

The fifth paper in the series [34] reiterates and extends the results found in [63] while taking a more geometric point of view. Once again the author considers a kinematic parameterization of the unitary propagators so that the particular form of the system Hamiltonian does not affect the analysis.

Consider the unitary group $\mathfrak{U}(N)$ and the group action $\mathcal{G} : \mathfrak{U}(N) \times \mathfrak{U}(N) \to \mathfrak{U}(N)$ defined by conjugation $\mathcal{G}(W, V) = WVW^\dagger$. The orbit of an element V under $\mathcal{G}$ is given by

$$\mathrm{Orb}_{\mathcal{G}}(V) = \left\{ \mathcal{G}(S, V) \;\middle|\; S \in \mathfrak{U}(N) \right\} \tag{3.12}$$

and the stabilizer of V is given by

$$\mathrm{Stab}_{\mathcal{G}}(V) = \left\{ S \in \mathfrak{U}(N) \;\middle|\; \mathcal{G}(S, V) = V \right\}. \tag{3.13}$$

The stabilizer and the orbit of a group element are related by the well known *Orbit-Stabilizer Theorem* from the theory of group actions[68]. It states that the orbit of an

element is isomorphic to the set of left cosets of the stabilizer, or in our case this can be more succinctly expressed as

$$\mathrm{Orb}_{\mathcal{G}}(V) \cong \mathfrak{U}(N)/\,\mathrm{Stab}_{\mathcal{G}}(V). \tag{3.14}$$

Let X_d be the desired unitary and consider the cost function as a measure of fidelity so that $C[V] = \Re\mathrm{Tr}[X_d^\dagger V]$. Noting that left-multiplication $X \mapsto X_d^\dagger X$ is an automorphism of the group, the landscape defined by the automorphism does not change. For this reason it is sufficient to replace $X_d^\dagger V$ by V and hence consider the cost function $C[V] = \Re\mathrm{Tr}(V)$. Our goal is to characterize the critical submanifolds of this cost function, and by [63] there are only a discrete set of critical values. Let $\{V_m\}$ describe the critical values, with an index ordering induced by the ordering of the values of $C[V_m]$. Using our arguments above and using the cyclic invariance of the trace, the critical submanifolds are $\mathrm{Orb}_{\mathcal{G}}(V_m)$ and hence are isomorphic to $\mathfrak{U}(N)/\,\mathrm{Stab}_{\mathcal{G}}(V_m)$. The authors then show that $\mathrm{Stab}_{\mathcal{G}}(V_m) \cong \mathfrak{U}(m) \times \mathfrak{U}(N-m)$ and hence the critical submanifolds are precisely the Grasmannian manifolds

$$\frac{\mathfrak{U}(N)}{\mathfrak{U}(m) \times \mathfrak{U}(N-m)}. \tag{3.15}$$

Of interest is that the authors then compute an abstract "volume" of these Grasmannians. We note that this abstract volume is only used to compare the relative size of volumes on the manifold and does not correspond to a physical volume. Nonetheless, it describes the relative size of these submanifolds with respect to all of $\mathfrak{U}(N)$. The authors are able to show that as N increases, these volumes decrease and play a less significant role in the landscape topology. Further, it should be clear that out-of-phase perfect controls $m = 0$ and in-phase perfect controls $m = N$ reduce the critical submanifolds to isolated points. The authors then move on to a familiar Hessian analysis which indicates that all other critical points correspond to saddle points.

Finally, [35] mimics much of the techniques and results of the previous paper, but with different constraint assumptions. We recognize that a common and important theme of the previous papers was using a kinematic point of view so as to avoid dealing with the specifics of the Hamiltonian. In particular, the only assumption made was that the Hamiltonian was Hermitian. This paper examines the landscape topology when the Hamiltonian is assumed to be real-symmetric or symplectic, which arise naturally when considering special symmetries in the system. The authors once again show that the only extrema correspond to prefect controls modulo a phase, and that all other critical points are non-optimal.

It is worth mentioning that the details of several papers originally listed have been omitted from this discussion, notably [33, 62, 64, 74]. While interesting and still pertaining

to quantum control, these papers tend to focus more on state transitions. Some of the techniques, justifications, and motivation are similar and may be of use to help understand the reasoning of the other landscape papers.

3.1.3 Finite Generation

Next we consider the finite generation of Lie groups, with particular attention paid to the Lie groups $\mathfrak{U}(N), \mathfrak{SU}(N)$ and how it can be directly exploited for the controlling quantum systems. These papers describe how we can "fill in the gaps" of a distribution using similarity transforms. For example, if G is a Lie group and $\mathcal{D}$ is a strict sub-bundle of TG, we can write elements of $TG \setminus \mathcal{D}$ in terms of exponentials of $\mathcal{D}$. Such similarity transforms are preserved by exponentiation and give a simple method of computing a control.

The paper [24] makes use of the structure theorem of Section 3.1.1 presented in [40], and adapts further results to the quantum control case. In particular let G be a Lie group and $\mathfrak{g}$ its Lie algebra. We recall that given a set $\{H_i\}_{i=0}^m$ which generates $\mathfrak{g}$, we can write any $X \in G$ as

$$X = \prod_{i=1}^{\ell} \exp[t_i H_{k_i}], \qquad k_i \in \{0, \ldots, m\}, t_i \in \mathbb{R}. \tag{3.16}$$

In particular, it can be shown that a uniform bound can be put on the order of generation ℓ so long as G is compact and the image of connected one-parameter subgroups of H_i are compact as submanifolds of G. The purpose of [24] is to relax the conditions on $\{H_i\}_{i=0}^m$ in a manner that will allow [25] to apply the results to quantum control theory.

Theorem 3.1.1. *Let G be a Lie group and $\{H_i\}_{i=0}^m$ denote a set of linearly independent elements of the Lie algebra $\mathfrak{g}$ which generate but do not necessarily span $\mathfrak{g}$. If the set does not span $\mathfrak{g}$, it is necessary that $\exists k, \ell$ with $k \neq \ell$ such that $[H_k, H_\ell] \neq 0$. Furthermore, there exists a $\hat{t}_{m+1} \in \mathbb{R}$ such that $\{H_i\}_{i=0}^m \cup \{H_{m+1}\}$ is linearly independent where*

$$H_{m+1} = \mathrm{Ad}_{\exp[\hat{t}_{m+1} H_\ell]} H_k. \tag{3.17}$$

The proof is simple and enlightening and so we include it here.

Proof. Since $\{H_i\}_{i=0}^m$ generate but do not span $\mathfrak{g}$, we know that there exists k, ℓ such that $[H_k, H_\ell] \neq 0$ and $\{H_i\}_{i=0}^m \cup \{[H_k, H_\ell]\}$ is linearly independent. Let $H_{m+1} = [H_k, H_\ell]$.

For the sake of contradiction, assume that for all $t \in \mathbb{R}$ we have that $\mathrm{Ad}_{\exp[tH_\ell]}(H_k)$ is linearly dependent on $\{H_i\}_{i=0}^m$. In particular, we can write H_{m+1} as a linear combination of the $\{H_i\}_{i=0}^m$, say

$$H_{m+1} = \sum_{j=0}^{m} a_j(t) H_j \tag{3.18}$$

where we have noted the explicit dependency of each a_j on the time parameter t. By applying the differential operator at $t = 0$, (3.17) yields

$$\begin{aligned}
\left.\frac{d}{dt}\right|_{t=0} H_{m+1} &= \left.\frac{d}{dt}\right|_{t=0} \mathrm{Ad}_{\exp[tH_\ell]}(H_k) \\
&= \Big[H_\ell \,\mathrm{Ad}_{\exp[tH_\ell]}(H_k) - \mathrm{Ad}_{\exp[tH_\ell]}(H_k) H_\ell\Big]_{t=0} \\
&= [H_\ell H_k - H_k H_\ell] \\
&= [H_\ell, H_k].
\end{aligned}$$

On the other hand, using the hypothesis of (3.18) we get

$$\begin{aligned}
\left.\frac{d}{dt}\right|_{t=0} H_{m+1} &= \left.\frac{d}{dt}\right|_{t=0} \sum_{j=0}^{m} a_j(t) H_j \\
&= \sum_{j=0}^{m} \frac{da_j}{dt}(0) H_j.
\end{aligned}$$

By equating our expressions above we find that

$$[H_\ell, H_k] = \sum_{j=0}^{m} \frac{da_j}{dt}(0) H_j \tag{3.19}$$

but since $\frac{da_j}{dt}(0)$ is simply a number, this implies $[H_\ell, H_k]$ is linearly dependent on $\{H_i\}_{i=0}^m$ which is a contradiction. □

By iteratively applying this technique, we can create an full basis $\mathcal{B} = \{H_i\}_{i=0}^{(\dim \mathcal{L})-1}$ for the Lie algebra. The procedure from here is a very simple one. Let X_d be the desired element and write this as $X_d = \exp[V]$ for some appropriate $V \in \mathfrak{g}$. Our goal is to apply the inverse function theorem[52] (IFT) in order to express X_d in terms of $\mathcal{B}$. By choosing $M \in \mathbb{N}$ sufficiently large we can ensure that $X_d^{\frac{1}{M}}$ is in a sufficiently small neighbourhood

of the group identity and apply the IFT to find that

$$X_d^{\frac{1}{M}} = \prod_{i=1}^{\ell} \exp[t_i X_{k_i}], \qquad k_i \in \{0, \ldots, (\dim \mathcal{L}) - 1\}, t_i \in \mathbb{R} \tag{3.20}$$

and we can write $X_d = (X_d^{\frac{1}{M}})^M$. The important step comes in noticing that this expression of X_d can then be translated to one exclusively using the original set $\{H_i\}_{i=0}^m$. In particular, we note that for $H_{m+1} = \mathrm{Ad}_{\exp[\hat{t}_{m+1} H_\ell]} H_k$ then

$$\exp[t_{m+1} H_{m+1}] = \exp[\hat{t}_{m+1} H_\ell] \exp[t_{m+1} H_k] \exp[-\hat{t}_{m+1} H_\ell]. \tag{3.21}$$

A similar procedure can be applied, recursively if necessary, to each H_k for $k > m$.

We can begin to see how this could be applied to open-loop quantum control theory. Our control fields are precisely the set $\{H_i\}_{i=0}^m$, and by applying this technique we can create any objective unitary in terms of exponentials of the control fields. This only works for driftless systems. Further, there is an obstacle in that the synthesis does not restrict the choice of t_i, so that we could in fact have negative propagation times. The second paper of D'Alessandro [25] deals with this issue. By considering the operator Schrödinger equation without drift and assuming controllability, the generating set $\{H_i\}_{i=0}^m$ is precisely the set of control fields. In order to deal with the problem of negative time, the author shows for if $\exp[-B|t|]$ is an element of the Lie group, then $\forall \epsilon > 0, \exists \bar{t} > 0$ such that

$$\left\| e^{-B|t|} - e^{B\bar{t}} \right\| < \epsilon \tag{3.22}$$

where $\|\cdot\|$ is the Frobenius norm. This says that the negative propagation of a flow can be arbitrarily approximated by some positive propagation.

However, if one attempts to implement this method computationally, a large slow-down occurs when trying to apply the inverse function theorem. Attempting to find the the appropriate decomposition near the identity results in a series of non-linear algebraic equations which can be both time consuming and difficult to solve. Consequently, the author provides an alternate method for arbitrarily precise control, and concludes the paper with an example of this technique on a low dimensional problem.

The use of similarity transformations to preserve the exponential is an interesting technique, and may be applicable to the implementation of numerical algorithms for solving time-optimal control. Unfortunately, while the results presented in these last two papers are interesting and pose a great deal of potential, they have yet to be fully exploited. A large part of this may be due to the lack of scalability of the methods provided above.

Indeed, since the dimension of two-level, n-particle spaces grows as 4^n, so too does the dimension of the Lie algebra. Numerically computing the full basis using the similarity transformation is akin to exhaustively exercising the LARC, but even more demanding as we need to retain information about which elements were used to create each new basis element.

3.1.4 Gradient Ascent Pulse Engineering

Let us conclude this section by considering numerical algorithms designed to construct quantum controls. In particular, we will examine the Gradient Ascent Pulse Engineering (GRAPE) algorithm presented by Khaneja et al. [48], which has received a great deal of attention and has been widely implemented. The purpose of this algorithm is to drive an initial state to a final state while maximizing the corresponding fidelity between the objective and the time-evolved state. While initially presented as a method for computing controls on density matrices, it can be generalized to unitary matrices using a small adaptation of the general technique. The derivation of this result given in [48] is only done in detail for control over density matrices. We will present a more thorough argument as it pertains to unitary evolution, though the argument is similar to that done for density matrices.

Consider the affine-controlled operator Schrödinger equation with drift

$$\frac{d}{dt}X(t) = -iH(t)X(t), \qquad H(t) = H_0 + \sum_{j=1}^{m} u_j(t)X_j \tag{3.23}$$

for $X(t) \in \mathfrak{U}(N)$ and $\{H_i\}_{i=0}^m \subset \mathfrak{u}(N)$. Define by $P(t)$ the propagator

$$P(t) = \mathcal{T} \exp\left[-i \int_0^t H(t)\ dt\right] \tag{3.24}$$

where $\mathcal{T}$ is the time-ordering operator, so that the solution to (3.23) is given by

$$X(t) = P(t)X(0). \tag{3.25}$$

Our goal is to derive an algorithm to drive the initial operator $X(0)$ to some target operator X_d in time T. For numerical reasons, let us then consider a uniform discretization of $[0, T]$ into M subintervals. It is not difficult to see then that the associated mesh space is precisely $\Delta t = T/M$ and the mesh points are given by $t_j = j\Delta t$ for $j = 0, \ldots, M$. We are

then content to find a solution to the control vector that is constant on each subinterval; that is, $\vec{u}(t_j) = (u_1(t_j), \ldots, u_m(t_j))$ for $t_{j-1} \leq t < t_j$.

Under this assumption, we can then approximate the continuous propagator $P(t)$ via a finite multiplicative set of time-independent propagators on each subinterval, $P(t) \approx P_M P_{M-1} \ldots P_2 P_1$. To discover the form of each P_i, we note that since the control Hamiltonians $\{H_i\}_{i=0}^m$ are time-independent and the control fields have been taken as constant on each subinterval $[t_{j-1}, t_j]$, we can find a propagator for fixed j given by

$$\begin{aligned} P_j &= \exp\left[-i \int_{t_j - \Delta t}^{t_j} \left(H_0 + \sum_{k=1}^{m} u_k(t_j) H_k\right) \, dt\right] \\ &= \exp\left[-i\Delta t \left(H_0 + \sum_{k=1}^{m} u_k(t_j) H_k\right)\right] \end{aligned}$$

Under this control action, the actual evolution of $X(0)$ in time T can be given by

$$X(T) = P_M \cdots P_1 X(0). \tag{3.26}$$

The cost function is the distance between the operators in the induced Frobenius norm,

$$\|X_d - X(T)\| = 2N - 2\Re \langle X_d | X(T) \rangle \tag{3.27}$$

so that minimizing $\|X_d - X(T)\|$ is equivalent to maximizing $C(X(T)) = \Re \langle X_d | X(T) \rangle$. By exploiting the definition of the adjoint operator, we notice that for any fixed $j \in \{1, \ldots, M\}$ we can write

$$C(X(T)) = \langle X_d | X(T) \rangle = \langle X_d | P_M \cdots P_1 X(0) \rangle \tag{3.28}$$

$$= \Big\langle \underbrace{P_{j+1}^\dagger \cdots P_M X_d}_{R_j} \Big| \underbrace{P_j \cdots P_1 X(0)}_{S_j} . \Big\rangle \tag{3.29}$$

Physically, R_j is the piece-wise constant backwards propagation of X_d from T to t_j, and S_j is the piece-wise constant forward propagation of $X(0)$ from 0 to t_j.

As we aim to implement gradient methods, we will need a way of determining a step direction in which to update. This is done by calculating the gradient of $C(X(T))$ with respect to each control function. It is important to note that $C(X(T))$ is implicitly a function of each $u_k(t)$ through the action of each propagator P_j, and by making each control field constant on subintervals of time we can consider $C(X(T))$ as a function of $m \times M$ variables u_{kj}, where $u_k(t) = u_{kj}$ on $[t_j, t_{j+1}]$. Hence our goal of finding an update

step is equivalent to computing $\frac{\partial C}{\partial u_{kj}}$. Let du_{kj} be an infinitesimal perturbation of u_{kj}, and denote the associated perturbation in P_j by $P_j + dP_j$. Since P_i is given by the exponential of a smooth function, it is necessarily smooth. We can deduce its derivative quite easily as

$$\frac{\partial P_j}{\partial u_{kj}} = \frac{\partial}{\partial u_{kj}} \exp\left[-i\Delta t\left(H_0 + \sum_{\ell=1}^{m} u_{\ell j} H_\ell\right)\right] \tag{3.30}$$

$$= -i\Delta t H_k P_j \tag{3.31}$$

This will allow us to conclude the essential part of the paper, which is that we can simplify the gradient calculation as

$$\frac{\partial C}{\partial u_{kj}} = \left\langle R_j | \frac{\partial P_j}{\partial u_{kj}} P_{j-1} \cdots P_1 X(0) \right\rangle \tag{3.32}$$

$$= \left\langle R_j | (-i\Delta t H_k P_j) P_{j-1} \cdots P_1 \rho(0) P_1^\dagger \cdot P_j \right\rangle \tag{3.33}$$

$$= -\left\langle R_j | i\Delta t H_k S_j \right\rangle \tag{3.34}$$

This gives us the direction of steepest ascent for the cost function with respect to each control, and motivates the following algorithm.

GRAPE Algorithm

1. Make an initial guess as to the control vector components u_{kj}.
2. For each time interval t_j, calculate the partial forward propagator $S_j = P_j \cdots P_1 X(0)$.
3. For each time interval t_j, calculate the partial backwards propagator $R_j = P_{j+1}^\dagger \cdots P_n^\dagger X_d$.
4. Compute $\frac{\partial C}{\partial u_{kj}}$, and update the control vectors according to $u_{kj} \to u_{kj} + \epsilon \frac{\partial C}{\partial u_{kj}}$ for some appropriately chosen stepsize ϵ.
5. If the change in $\frac{\partial C}{\partial u_{kj}}$ is less than a given tolerance, stop.
6. Return to step 2.

One can further improve upon this by utilizing a conjugate gradient scheme instead of merely steepest ascent [48].

This algorithm has been largely successful in computing controls, though one can see that it caters specifically to finding a control that optimizes the *fidelity*. In particular,

GRAPE was not designed for finding time-optimal controls, since the time length is initially fixed in order to discretize the control variables. We can attempt to approximate a time-optimal solution by continuously running this algorithm for successively smaller time lengths, stopping when the algorithm is no longer able to converge. This adaptation actually gives us a method for computing time-optimal controls that are not bang-bang, since the update scheme given in step 4 allows the controls to take values on the interior of their domain.

This thesis will analyze some methods of computational time-optimal algorithms in response to this void of appropriate methods, and a comparison to a modified *time-optimal* GRAPE will be made in Section 4.

3.2 Time-Optimal Control

Despite the advantages of considering time-optimal trajectories, this objective of quantum control has not received a great deal of attention, especially in the case of bounded control amplitudes. Indeed, most work has been done in considering the minimal time to access states under the hard-pulse assumption that the field magnitudes can be made arbitrarily large. Nonetheless, the geometric techniques explored in such a milieu are formidable and provide insight into the quantum landscape and as such are presented here. The similarity between sections here will allow us to go into more mathematical depth and provide a detailed examination of the results.

3.2.1 Coset Spaces

Our goal here will be to examine the work done by Khaneja et al. [45, 47] in calculating the minimal time for unitary transfer on an *unbounded* affine-control system. It is an application of the work done by Brockett [17] in geometric control theory to the domain of quantum mechanics. The focus of these papers is particularly limited to one- and two-qubit systems in order to take advantage of geodesic symmetry.

Let G be a Lie group with associated Lie algebra $\mathfrak{g}$. Consider the controllable system given by (3.23) where we allow the controls $u_j(t)$ to be unbounded. Intuitively, we will define an equivalence class of spaces in which we can move arbitrarily fast and then project onto an appropriately defined quotient space. The authors then show that the minimal time to move on this quotient space is, in fact, the minimal time to synthesize a unitary

operator. This stems from the fact that movement within an equivalence class can occur arbitrarily quickly and hence does not contribute to the overall time.

Consider first the Lie algebra $\mathfrak{k} = \text{Lie}\left(\{H_i\}_{i=0}^m\right)$, and let $K = \exp[\mathfrak{k}]$. The Killing Form $B : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ is defined by

$$B(X, Y) = \text{Tr}\left[\text{ad}_X \, \text{ad}_Y\right] \tag{3.35}$$

and is negative definite on compact Lie algebras. Since it is sign-definite, its negation can be used to define a Riemannian metric tensor on G, which is a non-degenerate, positive definite tensor field. This in turn can be used to define an inner product $\langle \cdot, \cdot \rangle_B$ on $\mathfrak{g}$. In fact, it can be shown that when G is either $\mathfrak{U}(N)$ or $\mathfrak{SU}(N)$ the negative Killing form is proportional to the standard trace inner product $\langle X, Y \rangle_{\text{Tr}} = \text{Tr}\left[X^\dagger Y\right]$.

We can use such an inner product to define the orthogonal complement of $\mathfrak{k}$ relative to $\mathfrak{g}$

$$\mathfrak{k}^\perp = \left\{ x \in \mathfrak{g} \;\middle|\; \langle x, y \rangle = 0, \forall y \in \mathfrak{k} \right\} \tag{3.36}$$

giving us the ability to decompose $\mathfrak{g}$ into orthogonal components $\mathfrak{g} = \mathfrak{k} \oplus \mathfrak{k}^\perp$. Such decompositions are not uncommon in Lie theory, and in the special case where the elements $\mathfrak{k}, \mathfrak{k}^\perp$ satisfy the relations

$$[\mathfrak{k}, \mathfrak{k}] \subset \mathfrak{k}, \qquad [\mathfrak{k}^\perp, \mathfrak{k}] \subset \mathfrak{k}^\perp, \qquad [\mathfrak{k}^\perp, \mathfrak{k}^\perp] \subset \mathfrak{k} \tag{3.37}$$

we say that $(\mathfrak{g}, \mathfrak{k})$ is a *Cartan decomposition* of $\mathfrak{g}$.

Now $K \leq G$ is a subgroup of the Lie group of G, and in particular we can define the group action of K on G by left-translation as

$$\mathcal{G} : K \times G \to G, \qquad (k, g) \mapsto kg \tag{3.38}$$

so that the orbit of $g \in G$ under K is just the left-coset Kg. From a control theoretical perspective, we note that by using unbounded controls we can move between any two elements of Kg arbitrarily quickly.

Now we want to consider how to examine the system modulo these spaces in which we can move arbitrarily quickly. Indeed, we recall that the quotient space G/K is a differentiable manifold (though not necessarily a Lie group unless $K \triangleleft G$ is normal). Furthermore, in the case where G/K is globally Riemannian symmetric, it follows that $\exp[\mathfrak{k}^\perp] = G/K$. The authors are then able to show that the adjoint action of the drift term H_0 is what generates the set of directions on this quotient space. More precisely, if $X \in K$ then in time $\Delta t, X$ evolves under the drift Hamiltonian as $\exp\left[-i\Delta t H_0\right] X$. This can be rewritten

as

$$\begin{aligned}\exp\left[-i\Delta t H_0\right] X &= X\left(X^\dagger \exp\left[-i\Delta t H_0\right] X\right) \\ &= X\left(\exp\left[-i\Delta t X^\dagger H_0 X\right]\right),\end{aligned}$$

and so evolution under the drift is an element of the coset given by the effective Hamiltonian $X^\dagger H_0 X$. Letting X vary through all K, we get the adjoint action of K on H_0, denoted $\mathrm{Ad}_K(H_0) \subseteq \mathfrak{k}^\perp$, which give the possible directions of movement in G/K.

In the case of time-optimal control, the authors argue that any time-optimal geodesic on G/K must have tangent vectors which always commute. Hence time-optimal trajectories will have their directions restricted to an abelian subalgebra $\mathfrak{h} \subseteq \mathfrak{k}^\perp$. In the event that $\mathfrak{h}$ is maximal amongst all abelian subalgebra, we say that $\mathfrak{h}$ is a *Cartan subalgebra*. Important to this study and section 3.2.2 is the fact that if $\mathfrak{h}$ is a Cartan subalgebra of a Cartan decomposition $(\mathfrak{g}, \mathfrak{k})$ then we can write $G = K \exp\left[\mathfrak{h}\right] K$.

With the mathematics established, we want to specialize our consideration to time-optimal synthesis. Let $\mathcal{A}(X, T)$ denote the accessibility set of $X \in G$ in time *at most* $T > 0$. Then $\forall Y \in A(X, T)$ let $\hat{t}(Y)$ denote the minimal time to drive X to Y. Next the authors define what they call an "adjoint control system" on G given by

$$\frac{d}{dt} P(t) = X P(t), \qquad X \in \mathrm{Ad}_K(X_d). \tag{3.39}$$

We advise the reader to be wary of this terminology, as there is an alternative notion of adjoint systems in control theory which does not agree with this definition. Define the set $B(X, T)$ to be the set $K \cdot R(X, T)$; that is, the orbit of the reachable set of X under K. We define $\forall Y \in B(X, Y)$ the number $\tilde{t}(Y)$ to be the minimal time to drive X to Y. Alternatively, if $Y \in R(X, T)$ then $\tilde{t}(Y)$ is the minimal time to drive X to the coset KY.

Let $I \in G$ denote the identity element. The main result of this paper is that given a desired unitary $X_d \in R(I, T)$, the minimal time $\tilde{t}(X_d)$ to drive I to the coset KX_d as governed by (3.39) is identical to the minimal time $\hat{t}(X_d)$ to drive I directly to X_d under (3.23). Hereafter we are justified to move to a quotient space of cosets in order to calculate the infimum time to synthesize X_d.

The general techniques of [45] are then directly applied to particular unitary operators in [47]. Combined with the PMP, the authors consider a specific class of operators important to the study of nuclear magnetic resonance (NMR) and establishes the time-optimal solutions for a three-spin system.

These papers work to exploit the differential geometric nature of the control manifold to establish results regarding time-optimal trajectories, applying many results of Cartan's

work combined with the assumption of Riemannian symmetry. Unfortunately, the symmetric assumption very quickly breaks down for more than two-body systems, and necessitates an alternate treatment presented in the following section.

3.2.2 Cartan Decompositions

Using the notion of Cartan decompositions introduced in [45], this section hopes to illuminate the results of [46]. The Euler-angle decomposition of $\mathfrak{SO}(3)$ and consequently $\mathfrak{SU}(2)$ are well known and give the ability express any element of the aforementioned Lie groups in terms of simple rotations. The purpose of [46] is to extend this idea to $\mathfrak{SU}(2^n)$ so that arbitrary n-qubit operators can be recursively decomposed into two-qubit gates.

The Euler angle decomposition of $\mathfrak{SU}(2)$ states that, given the Pauli basis $\{I_x, I_y, I_z\}$

$$I_x = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{3.40}$$

$$I_y = \frac{1}{2}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{3.41}$$

$$I_z = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{3.42}$$

we can write any $X \in \mathfrak{SU}(2)$ as

$$X = \exp[-i\theta_x I_x] \exp[-i\theta_y I_y] \exp[-i\theta_z I_z]. \tag{3.43}$$

Again let G be a Lie group, $\mathfrak{g}$ its Lie algebra, and controllable system dynamics given by (3.23) with $u_j(t)$ unbounded. The work of [45] was to show that for a globally Riemannian symmetric Lie group G with a Cartan decomposition $\mathfrak{g} = \mathfrak{k} \oplus \mathfrak{k}^\perp$ and a Cartan subalgebra $\mathfrak{g} \subseteq \mathfrak{k}^\perp$ containing H_0, we can write $G = K \exp[\mathfrak{h}]K$. We will now turn our attention to the specific case where $G = \mathfrak{SU}(2^n)$ is not necessarily Riemannian symmetric and give a similar decomposition.

For brevity, consider $\mathfrak{SU}(2^n)$ for some $n > 2$ and keep the same notation as that given in Section 3.2.1. The authors are able to show that we can decompose $\mathfrak{SU}(2^n)$ as

$$\mathfrak{SU}(2^{n-1}) \otimes \mathfrak{U}(1) \subset \mathfrak{SU}(2^{n-1}) \otimes \mathfrak{SU}(2) \subset \mathfrak{SU}(2^n). \tag{3.44}$$

In particular, define

$$\mathfrak{su}_{\mathfrak{k}}(2^n) = \operatorname{span}\left\{ A \otimes I_z, B \otimes \mathrm{id}, iI_{nz} \;\middle|\; A, B \in \mathfrak{su}(2^{n-1}) \right\} \tag{3.45}$$

$$\mathfrak{su}_{\mathfrak{k}0}(2^n) = \operatorname{span}\left\{ A \otimes I_x, iI_{nz} \;\middle|\; A \in \mathfrak{su}(2^{n-1}) \right\}. \tag{3.46}$$

If we let $\mathfrak{h}$ be a Cartan subalgebra of $(\mathfrak{su}(2^n), \mathfrak{su}_k(2^n))$ and $\mathfrak{f}$ be a Cartan subalgebra of $(\mathfrak{su}_k(2^n), \mathfrak{su}_{k0}(2^n))$ then for any $U \in \mathfrak{SU}(2^n)$ we can write

$$U = K_1 \exp[Z_1] K_2 \exp[Y] K_3 \exp[Z_2] K_4 \tag{3.47}$$

where $K_1, K_2, K_3, K_4 \in \mathfrak{SU}(2^{n-1}) \otimes \mathfrak{U}(1)$, $Y \in \mathfrak{h}$, and $Z_1, Z_2 \in \mathfrak{f}$.

We first notice that while this may be useful from a control theoretic point, as it stands we cannot directly apply it to any problem. In particular, the choice of the Cartan subalgebra needs to be specified by the Hamiltonian drift, and we cannot be guaranteed that our control fields generate a Cartan decomposition. Nonetheless, this may provide some insight into the internal mechanics of certain systems, and provide a tool for computation when the aforementioned caveats are satisfied.

It can also be quite technical to compute some of the desired quantities listed above, especially as the size of the system grows exponentially in the number of qubits: given an n-body system, $\mathfrak{SU}(2^n)$ has dimension $4^n - 1$. There has been some work to create constructive algorithms for the Cartan decomposition using this scheme as motivation, and in particular we refer the reader to [27] for more information. The details are quite technical and do not promise to scale well with the number of qubits, but give a starting point for computation.

3.2.3 Other Results

Herein we briefly mention other results in time-optimal control. In all cases, the papers mentioned below are either beyond the scope of this thesis, are computationally intensive without providing a great deal of theoretical insight, or present results for which only indirect manipulation of solutions yields relevant information.

In a fascinating but very complex paper by Agrachev and Chambrion [4], the issue of controllability for general compact Lie groups is considered. In particular, the case wherein the Lie algebra of control fields is not a part of the Cartan decomposition is considered. The authors restrict their attention to single-input systems of dimension greater than three. Such systems cannot be part of the Cartan decomposition for dimensional reasons. The paper uses many of the ideas presented above in its formulation, especially the root-space decomposition technique introduced in section 3.1.1. The treatment is done for general compact Lie groups though the authors make specific mention of the intent to treat optimal quantum control objectives.

The paper [16] examines time minimal trajectories of a single particle in a magnetic field, abstracted as a single-input affine system with normalized, bounded controls. While

many other papers assume that the drift Hamiltonian can be eliminated, this paper deals specifically in the instance where it is necessarily present. The authors consider the problem comprehensively and are able to resolve the spin-$\frac{1}{2}$ case in its entirety. This is done by breaking the Bloch sphere into eight open regions and specifying the control sequences dependent on the region in which the initial state lies. The authors are able to show that in some instances, not all solutions are bang-bang and some must indeed be bang-singular.

Despite the paper's focus on optimal state transition, we include it here since it can be connected to unitary synthesis. For example, two-level systems occupy the 3-sphere $S^3 \subseteq \mathbb{C}^2$, which is diffeomorphic to $\mathfrak{SU}(2)$. Hence by associating desired states on S^3 to operators on $\mathfrak{SU}(2)$, the optimal trajectories between elements of S^3 can be diffeomorphically mapped into $\mathfrak{SU}(2)$. One possible caveat is that most considerations of two-level systems are projected from the three-sphere onto the Bloch sphere, and this paper is no exception.

In [76] the authors make use of theory of theoretical considerations described in [46] to compute unitary transformations important to electron paramagnetic resonance experiments wherein one qubit couples significantly faster than the other. Specifically, the authors use Cartan decompositions to construct the time optimal synthesis of operators on $\mathfrak{SU}(4)$. The paper is dedicated to computation based off of the work of Khaneja and Glaser, and provides results for very specific objective unitaries. However, it does not further add to the theory of time-optimal controls and so we have omitted any further discussion.

Finally, [13, 15, 51] consider time-optimal control of two-level dissipative systems and, in particular, the existence of singular extremals. In [51], the authors are even able to show that singular controls allow for a more optimal solution than the best known bang-bang control. Inversion recovery is the intuitive technique used in NMR and is a bang-drift trajectory, whereas it is shown that a bang-singular hybrid control yields better implementation times.

Having discussed the topic of control theory in general, we are now in a position to examine the more specific goals of this thesis. The next section shall introduce numerical algorithms whose purpose is to compute time-optimal controls. This will be followed by an examination of conditions for singular controls.

Chapter 4

Numerics

We are driven to consider numerical algorithms for several reasons. One possibility is that analytical solutions are often very specific to the systems in which they are applied. Changing the objective unitary, the drift term, or even one of the control fields may necessitate an entirely different analysis than similar setups. Another example is that some systems fail to admit analytical solutions, or the solution may be too complicated to justify computing. In these cases, we often defer to numerical computations to give results and approximations.

One should not depend entirely on numerical results though, as analytical solutions can often reveal subtle properties that numerics cannot always distinguish. We mention a particular instance where the analytical solution to an optimal control problem exceeded the computational solution. In [65], Reeds and Shepp consider the optimal-length trajectories for manipulating a car using "an impressive array of techniques especially developed for the study of this particular problem," including "the use of [a] computer [which] played a fundamental role." In a follow up by Sussman and Tang [72], the authors were able to show that the intricate and complex solution posed by Reeds and Shepp could actually be succinctly solved using geometric control theory. Furthermore, Sussman and Tang go on to show that while Reeds and Shepp were able to describe optimal trajectories in terms of the concatenation of 48 three-parameter families, the mathematically optimal procedure only requires 46 such families and hence provides a more efficient characterization of the controls through analytic treatment.

Nonetheless, the need for convergent and practical numerical algorithms cannot be overlooked. It has been mentioned in Chapter 3.1.4 that there is a particular void of algorithms that compute time optimal controls. This chapter hopes to provide an overview

of some of the work that has been done on such algorithms, as well as the our own work with regards to the topic.

4.1 A Review of GRAPE

We begin our treatment by recalling Khaneja's work on computing high-fidelity control laws via the GRAPE algorithm, which was covered in detail in Section 3.1.4. We refer the reader to that section of this thesis as well as [48] for more details, though we re-iterate the theory below.

The general procedure of GRAPE is to use a steepest ascent or conjugate gradient method to optimize a cost function which measures the similarity between an objective and the current state. In the case of unitary operators, this reduces to the fidelity measure (3.27) using the Hilbert-Schmidt inner product. In particular, Khaneja examines how this fidelity will change with respect to small variations in a discretized set of controls and derives an algorithm based on gradient ascent. This technique is then used to update the control fields until a sufficient tolerance is reached.

We note that the primary result of the paper was to re-express the component-wise gradient calculation in terms of a much simpler expression. Explicitly, if C denotes the cost function, u_{kj} is the k^{th} control on the j^{th} time interval, and the final state $X(t)$ is given by $X(t) = P_M \cdots P_1 X(0)$ we can write

$$\frac{\partial C}{\partial u_{kj}} = -\left\langle P_{j+1}^{\dagger} \cdots P_M X_d | i\Delta t H_k P_j \cdots P_1 X(0) \right\rangle.$$

This is very useful since derivative calculations are computationally expensive and unstable actions. Naive approximations to differentials are done by subtracting two similar results and dividing by a small number, both of which are operations that can substantially contribute to error proliferation[18]. This update scheme avoids both of these problems and provides an expression of the gradient in terms of quantities that are already known.

However, as mentioned in Section 3.1.4 this algorithm was not designed to deal with time-optimal calculations. Since the terminal time is fixed via discretization, any attempt to find time-optimal trajectories is the result of repeated application of the algorithm to successively shorter time intervals. While the number of ways to successively shorten such an interval are many, for our comparison we have implemented a bisection method by which the time interval is halved at each iteration. Once the algorithm no longer converges, the user assumes that this is because no smaller solution can be found and terminates the search.

4.2 The Kaya-Huneault Algorithm

4.2.1 Kaya-Huneault Method

The absence of explicit methods for time-optimal control heavily motivated the thesis [37], which worked to adapt an existing algorithm from classical control into a quantum mechanical framework. It is from this body that the foundation of our own work is built, and hence we offer a detailed consideration of the motivation and techniques that it introduces.

We quickly re-iterate the problem as stated in [37] so as to avoid any ambiguity that might arise from the notation. Recall that our principle goal is to find an efficient way of computing time-optimal controls for synthesizing unitary gates on quantum mechanical systems. Let $d \in \mathbb{N}$ represent the dimension of our system and let $X \in \mathfrak{U}(d)$ be an element of unitary space with the condition $X_0 = X(t_0)$ at the time t_0. If the energy Hamiltonian of the system is $H \in i\mathfrak{u}(d)$, we can expect X to evolve according to the Schrödinger equation

$$\hbar \frac{d}{dt} X(t) = -iHX(t), \qquad X(t_0) = X_0. \tag{4.1}$$

Carrying around the $\hbar$ term can by quite cumbersome and so we often choose to normalize our units so that $\hbar \equiv 1$. After any solution is computed, for it to be physically interpreted we must make appropriate transformations to revert to the standard value of $\hbar$, but there is no loss in generality in working with this simplified picture.

We choose to impose our presence on the system by affecting the nature of the energy Hamiltonian directly. Assume that we have an m-input system for some $m \in \mathbb{N}$, writing H as

$$H(t) = H_0 + \sum_{i=1}^{m} u_i(t) H_i, \tag{4.2}$$

where H_0 is the drift Hamiltonian and represents evolution in the absence of control, the H_i are fixed Hamiltonians representing the effects of our physical influence over the system, and the $u_i(t)$ represent the magnitude of each control Hamiltonian whose optimal values are the primary objective of this treatise.

We have assumed that we are given an initial starting point $X(t_0) = X_0$ and hope to drive the unitary operator to a desired state $X_d \in \mathfrak{U}(d)$ in a time-optimal way. However, we need a way of measuring how "close" we are to our objective X_d. The notion of measuring distances in a space requires the definition of a metric, or for our purposes a function $d : \mathfrak{U}(d) \times \mathfrak{U}(d) \to \mathbb{R}$ acting on unitary elements. As unitary space is well known to be a complex Hilbert space we can exploit the inner product to induce a norm, which in turn

induces a metric. If our Hilbert space is $(\mathcal{H}, \langle \cdot, \cdot \rangle)$, then the inner product gives the metric as

$$\langle X_1 - X_2, X_1 - X_2 \rangle = \|X_1 - X_2\|^2 = d(X_1, X_2)^2. \tag{4.3}$$

We have already seen a way of simplifying this expression, given in Proposition 2.4.8 as

$$d(X_1, X_2)^2 = 2d - 2\Re\mathrm{Tr}[X_1^\dagger X_2]. \tag{4.4}$$

Since a metric is always non-negative and non-degenerate, minimizing $d(X_1, X_2)^2$ is equivalent to minimizing $d(X_1, X_2)$ and so we will often consider the squared-metric in order to avoid messy square-roots.

We notice here a relationship between our metric and the gate fidelity $F(X_1, X_2) = 2^{-d}\mathrm{Tr}[X_1^\dagger X_2]$. This is perhaps unsurprising as fidelity measures how well two gates overlap (their closeness) whereas our metric measures how far they are apart (their distance). With this metric at our disposal and a well-defined objective unitary X_d, we can define an objective function $f : \mathfrak{U}(d) \to \mathbb{R}$ by $f(X) = d(X, X_d)^2$.

The PMP indicates that since our system is affine in the bounded control variables, any non-singular time-optimal solutions should be of a bang-bang nature. The additional assumption that our Hamiltonian is given by (4.2) allows us to apply a normalization condition. If the control $u_i(t)$ corresponding to the control amplitude of H_i is restricted to lie in the set $u_i(t) \in [0, M]$, for some $M > 0$, we can redefine $\tilde{H}_i = MH_i$ so that $\tilde{u}_i(t) \in [0, 1]$ results in the same range of control. Without loss of generality, we will henceforth assume that all controls are normalized. The assumption that there are no singular trajectories then indicates that each control occupies the boundary of its domain and hence is either at full capacity or off during a given time interval; mathematically, this corresponds to $u_i(t) \in \{0, 1\}$, for each $i = 1, \ldots, m$.

With the theoretical background established, we can move to casting this into a computational framework. Some of the procedures will look similar to those introduced by the GRAPE algorithm in section 3.1.4, which occurs because these are standard discretization techniques.

The form of (4.1) and its similarity to the exponential yields an analytic solution as follows:

$$X(t) = \mathcal{T} \exp\left[-i \int_{t_0}^{t} \left(H_0 + \sum_{i=1}^{m} u_i(t) H_i\right) \, dt\right] \tag{4.5}$$

where $\mathcal{T}$ is the time-ordering operator. If T is the total time required to drive X_0 to X_d we can partition the interval $[0, T]$ as $0 = \tau_0 < \tau_1 < \ldots < \tau_{\ell-1} < \tau_l = T$ such that

$H_0 + \sum_{i=1}^{m} u_i(t) H_i$ is constant on each interval $\tau_{j-1} < t < \tau_j$ for $j = 1, \ldots, \ell$. For simplicity of notation, define

$$\tilde{H}_j = H_0 + \sum_{i=1}^{m} u_i(t) H_i \Bigg|_{\tau_{j-1} < t < \tau_j} \qquad \xi_j = \tau_j - \tau_{j-1}, \qquad j = 1, \ldots, m. \tag{4.6}$$

This definition implies that $\tilde{H}_j$ is the effective Hamiltonian when applied over the interval (τ_{j-1}, τ_j), which is constant and is in effect for a length ξ_j. Since each component of $\tilde{H}_i$ is in $i\mathfrak{u}(d)$, it follows that $\tilde{H}_i \in i\mathfrak{u}(d)$. This allows us to remove the time-ordering from (4.5) and parameterize $X(t)$ in terms of ξ as

$$X(\xi) = \exp\left[-i\tilde{H}_\ell \xi_\ell\right] \exp\left[-i\tilde{H}_{\ell-1}\xi_{\ell-1}\right] \cdots \exp\left[-i\tilde{H}_2\xi_2\right] \exp\left[-i\tilde{H}_1\xi_1\right] \tag{4.7}$$

$$= \prod_{i=0}^{\ell-1} \exp\left[-i\tilde{H}_{\ell-i}\xi_{\ell-i}\right]. \tag{4.8}$$

We would like to point out that we have assumed the convention that multiplication always occurs to the right, and so our choice of indexing is to emphasize the decreasing order in which the product is to be taken. The definition of ξ_i makes it clear that $\xi_i \in \mathbb{R}$ and $\xi_i \geq 0$, for each $i = 1, \ldots, m$. If we are given a collection of effective Hamiltonians $(\tilde{H}_1, \ldots, \tilde{H}_\ell) \in (i\mathfrak{u}(d))^\ell$ and a vector $\vec{\xi} = (\xi_1, \ldots, \xi_\ell) \in \mathbb{R}_+^\ell$, we can define an approximate evolution function $g_\ell : \mathbb{R}_+^\ell \times (i\mathfrak{u}(d))^\ell \to \mathfrak{U}(d)$ that acts on the discretization by creating a forward propagation operator

$$g_\ell\left(\vec{\xi}, (\tilde{H}_1, \ldots, \tilde{H}_\ell)\right) = \prod_{i=0}^{\ell-1} \exp\left[-i\tilde{H}_{\ell-i}\xi_{\ell-i}\right]. \tag{4.9}$$

We are now in a position to introduce the motivation and techniques used in [37]. The following algorithm is very complicated and uses subtle assumptions that have yet to be verified. These assumptions and their possible shortcomings are discussed below in Section 4.2.2. The original coded implementation contains errors and inefficient subroutines that are a direct result of the intricacy of this algorithm. Despite the fact that much of our work was to make it less complicated and theoretically sound, we nonetheless recapitulate the original presentation below.

Kaya et al. have published a series of papers on implementing algorithms to calculate time-optimal controls for non-linear systems [41, 42, 43]. The technique introduced in

these algorithms is to reduce the problem to the subspace of all possible control techniques that correctly implement the desired objective and then to minimize the total time of implementation using a linearize-and-project technique. The work of Huneault in [37] was to specialize this algorithm to unitary evolution so as to be applicable to quantum mechanics, as well as to extend the result to a multiple-control regime.

Recall that if $X_d \in \mathfrak{U}(d)$ is our desired output, then $f : \mathfrak{U}(d) \to \mathbb{R}_+$ is given by $f(X) = d(X, X_d)^2$. Fix an $\ell \in \mathbb{N}$ and a control set $(\tilde{H}_1, \ldots, \tilde{H}_\ell) \in (i\mathfrak{u}(d))^\ell$. In order to consider the subspace of $\mathbb{R}^l$ which generates X_d under this control set, we note that we can write $X(\xi)$ in (4.8) using (4.9) as $X(\xi) = g(\xi)X_0$. Define $S : \mathbb{R}^\ell \to M_n(\mathbb{C})$ as $S(\xi) = X(\xi) - X_d$ so that the preimage of zero gives us a subspace which corresponds to switching times that drive X_0 to X_d. We can be more precise by defining

$$\Omega = S^{-1}(\tilde{0}) \tag{4.10}$$

where $\tilde{0}$ is the zero matrix in $M_n(\mathbb{C})$. We refer to Ω as the *terminal surface*. Our goal is to minimize our implementation time while constraining ourselves to this space. Since we have parameterized our controls in terms of switching intervals ξ, the total time of any given implementation is then given by

$$t_f(\xi) = \sum_{i=1}^{\ell} \xi_i \tag{4.11}$$

which can alternatively be written as

$$t_f(\xi) = c^T \xi, \qquad c^T = \underbrace{(1, \ldots, 1)}_{\ell \text{times}}. \tag{4.12}$$

In linear programming, the objective function is given by $f : \mathbb{R}^n \to \mathbb{R}$ where $f(x) = c^T x$ for some constant vector $c \in \mathbb{R}^n$. Comparing this to (4.12) we see that $t_f(\xi)$ is an objective function for a linear programming problem. However, the condition that $S(\xi) = 0$ yields a non-linear constraint. In order to apply the techniques of linear programming, we consider the tangent space at a point $\xi_0 \in \Omega$. This creates a linear space on which a linear programming problem can then be solved. In particular, we change our constraint set to

$$\nabla S(\xi_0) \cdot (\xi - \xi_0) = 0 \tag{4.13}$$

resulting in the linear programming problem

$$\begin{aligned} &\min_{x \in \mathbb{R}^\ell_+} c^T \xi \qquad \text{subject to} \\ &\nabla S(\xi_0) \cdot (\xi - \xi_0) = 0. \end{aligned} \tag{4.14}$$

Let ξ_{LP} denote the solution to (4.14), which need not lie in Ω and hence need not correspond to a desirable control law. After solving for ξ_{LP}, we must then project this solution back onto Ω. If the projection is successful, we re-iterate the process until a minimal change in the time is seen. If we are unable to project back onto the surface or the projection does not yield a better time, the midpoint between ξ_0 and ξ_{LP} is calculated and the projection step is attempted again. After the procedure has been completed, one removes zero-length arcs and merges arcs corresponding to identical controls.

To generalize this algorithm to the case of multiple controls, the author introduces a series of constructs to keep track of which controls switch on a given interval. Given k switching times $\{\tau_i\}_{i=1}^{k}$, we can discretize the interval $[0, T]$ as $0 = \tau_0 < \tau_1 < \ldots < \tau_k = T$. Define the following:

- $\mathcal{C}$: A vector in $\mathbb{N}^k$ whose i^{th} coordinate is the value j corresponding to which control $u_j(t)$ switches at time τ_i.
- α^* : A matrix of dimension at most $k \times m$, whose element α^*_{ij} is the length between the i^{th} and $(i+1)^{th}$ switch of the j^{th} control, or zero if the index exceeds the number of switches for the j^{th} control.
- σ : A vectorization of α^* that omits the out-of-index zeros.

The variables α and σ contain sufficient information to determine the evolution of our initial operator in precisely the same manner as ξ. Consequently, we can consider a reparameterization of $X(\xi)$ as $\hat{X}(\sigma)$ or $\tilde{X}(\alpha)$ which gives us the ability to consider several different spaces over which we might choose to optimize. Huneault is able to construct a transformation operator B that permits movement between the spaces designated by the α, ξ, and σ representations, with $\xi = B(\sigma, C)$ and $\sigma = B^{-1}(\xi, C)$. We omit the details of the derivation of this transformation here, but we state that it is easily calculated by considering the generalized Moore-Penrose inverse between the vector spaces in which ξ and σ occupy. The constraint space of the linear programming problem is then converted from ξ space to σ space by reparameterizing the terminal surface as

$$S(\sigma) = X(B(\sigma, C)) - X_d = 0. \tag{4.15}$$

Since $\xi = B(\sigma, C)$, equation (4.15) and the constraints given by (4.10) yield equivalent representations of the terminal surface Ω. However, this change of variables necessitates a change in the objective function of the linear programming problem. While we had previously considered the sum over each component of ξ, the summation over all σ yields

$$\sum_{i=1}^{\dim \sigma} \sigma_j = m \times t_f(\xi). \tag{4.16}$$

This occurs because σ contains the complete set of switching intervals for each of the m inputs. To ensure that we are adhering to the same minimization problem as that given by (4.14), we need only extract the total time from a single control set since we demand that all control intervals last precisely the same amount of time. To do this, consider the first control $u_1(t)$ which we declare to have κ_1 switching intervals. The corresponding change of parameterization for $t_f(\xi)$ as $t_f(\sigma)$ is given by by

$$t_f(\sigma) = \hat{c}^T\sigma, \qquad \hat{c}^T = (\underbrace{1,\ldots,1}_{\kappa_1\,\text{times}},0\ldots,0). \tag{4.17}$$

This choice of $\hat{c}$ selectively withdraws only the information about the interval length for the first control. However, since the dimension of σ is guaranteed to be at least as large as ξ, our constraints given by (4.15) are under-determined. This can be resolved by realizing that in defining $\hat{c}$, we discarded information about all controls other than the first. Imposing the additional equality constraints that each control have the same duration, we can prescribe a properly determined system.

With this theory established, we present below a summary of the algorithm:

Let $X_d \in \mathfrak{U}(d)$ be the objective unitary transformation. Let $\ell \in \mathbb{N}$ describe the number of discretized intervals, and prescribe a set $(\tilde{H}_1, \ldots, \tilde{H}_\ell)$ of time-independent effective Hamiltonians on each interval.

1. Find a point on the terminal surface Ω. Equivalently, calculate an initial set of switching intervals ξ_0 that drives the initial point X_0 to X_d.
2. Linearize the terminal surface Ω about the point ξ_0 and solve the linear programming problem given by (4.12) in the case of a single-input system, or (4.17) in the multiple-input case. Let ξ_{LP} be this solution. If $\xi_0 = \xi_{\text{LP}}$ then go to step 5.
3. Given ξ_{LP}, attempt to project back to the terminal surface. If it is not possible to project back to Ω or the projection yields a greater implementation time, find the midpoint between ξ_0 and ξ_{LP} and try again.
4. If no appropriate point can be found, terminate the algorithm. Otherwise, set ξ_0 to be the projection point and go to step 2.
5. If no intervals have collapsed we append a zero-length intervals to the beginning and end of the control sequence and go to step 2.
6. Remove collapsed intervals and merge similar intervals as necessary. Terminate the algorithm.

4.2.2 Criticism

The actual implementation of this algorithm initially met with very limited success, essentially only providing feasible results in the two-level single-particle case. This is the result of the algorithm being based on subtle and unverifiable assumptions discussed below.

We begin by recalling the definition of the terminal surface, given in (4.10) as $\Omega = S^{-1}(\tilde{0})$ where $S : \mathbb{R}^\ell \to M_d(\mathbb{C})$. Since $\tilde{0}$ represents the zero-matrix in $M_d(\mathbb{C})$, calculating the pre-image can be difficult and time consuming. Indeed, numerically we make no attempt to calculate the pre-image and instead satisfy ourselves to find a single point in Ω. Furthermore, since S is a matrix valued function its gradient will be a 3-tensor. We will need to be careful when evaluating the vector contraction to ensure that we are dimensionally consistent. We provide here a more rigorous treatment for calculating the terminal surface than that provided in [37].

Choose an arbitrary $\xi \in \mathbb{R}^\ell$ and let $X = S(\xi) \in M_d(\mathbb{C})$. In particular, we choose a tensorial representation of X as $X = X^i_j(\xi) e_i \otimes e^j$ where $X^i_j(\xi) \in \mathbb{C}$ and $\{e^j\}$ is the standard basis in $\mathbb{R}^d$ with dual basis $\{e_i\}$. To apply a "gradient" to X, we consider the operator $\nabla = \frac{\partial}{\partial \xi^k} e^k$, and apply it to X to get

$$\nabla X = X^i_{j,k}(\xi) e_i \otimes e^j \otimes e^k, \qquad X^i_{j,k} = \frac{\partial X^i_j(\xi)}{\partial \xi^k}. \tag{4.18}$$

Our goal now is to contract ∇X with the vector $\xi = \xi^\kappa e_\kappa$, which can be written as

$$\begin{aligned}(\nabla X)\xi &= X^i_{j,k} \xi^\kappa e_i \otimes e^j \otimes e^k(e_\kappa) \\ &= X^i_{j,k} \xi^k e_i \otimes e^j.\end{aligned}$$

It is important to note here that while $X^i_{j,k}$ contains two covariant components over which a contraction could occur, only the k-component can actually admit a proper reduction in the rank of the tensor since the dimension of the j component generally disagrees with the k component.

Linearizing the terminal surface requires the computation of $X^i_{j,k}$ and then performing the contraction. Needless to say, calculating $(1,2)$-tensors is costly and can impair the speed and efficiency of a numerical algorithm. This is further exacerbated by the fact that linear programming problems are done in real space requiring that the $(1,1)$-tensor $X^i_{j,k}\xi^k$ be transformed from a d^2-dimensional complex tensor to a $2d^2$-dimensional real tensor. The transformation from complex $(1,1)$-tensors into $(1,1)$-real tensors is something that is normally requires an embedding of d^2-dimensions into $4d^2$-dimensional space[8], but as our goal does not require the preservation of the ring-structure of $M_d(\mathbb{C})$ it suffices to identify $\mathbb{C} \cong \mathbb{R}^2$ giving the desired embedding.

In the context of the previous calculation, the derivative of the evolution mapping $X(\xi)$ as given by (4.7) is particularly simple. By writing

$$X(\xi) = \exp\left[-i\tilde{H}_\ell \xi_\ell\right] \exp\left[-i\tilde{H}_{\ell-1}\xi_{\ell-1}\right] \cdots \exp\left[-i\tilde{H}_2\xi_2\right] \exp\left[-i\tilde{H}_1\xi_1\right] \tag{4.19}$$

the partial derivative $\frac{\partial X(\xi)}{\partial \xi_k}$ is given by

$$\frac{\partial X(\xi)}{\partial \xi_k} = \left(\prod_{i=0}^{\ell-k-1} \exp\left[-i\tilde{H}_{\ell-i}\xi_{\ell-i}\right]\right)(-i\tilde{H}_k)\left(\prod_{i=0}^{k-1} \exp\left[-i\tilde{H}_{k-i}\xi_{k-i}\right]\right). \tag{4.20}$$

The projection algorithm is assumed given in [41] and is not greatly elaborated upon in [37], wherein the author uses a conjugate gradient method. However, this projection algorithm turns out to be one of the primary obstacles in implementing the Kaya-Huneault algorithm. We will see that the projection step is especially ill-suited when we consider more general topological issues in Section 4.3.3. It is suggested in [37] that one should search in orthogonal directions relative to the tangent hyperplane generated by the linearization, though more work on an analytic expression of such directions is needed. This evidence against projection techniques seems to suggest that any algorithm which requires a projection step may be infeasible.

Finally, we comment on the unnecessary complexity involved in the transfer between σ and ξ spaces which only proves to make the problem more abstruse. In order to track the pertinent information regarding the switches, the author created the variables α, α^*, σ, and $\mathcal{C}$ which are all defined on page 69 above, as well as another variable κ whose elements track the number of switchings for each control. As each control may have a different number of unique switches, one often needs to pad matrices with extra zeros in order to be dimensionally consistent. The constant addition and subtraction of unnecessary zeroes requires cumbersome and time consuming checks in order to avoid dimension mismatch errors in vector calculations. These issues propagate throughout the algorithm resulting in an overly complicated coding scheme.

The variables α, α^*, σ and κ carry a large amount of redundant information and results in an excessive amount of accounting when implemented. This can be mitigated by maintaining a array corresponding to the logical state of each control during each switching interval; something that is done in the Kaya-Huneault code anyway.

The inability of this algorithm to work for even small numbers of particles may have simply been a result of theoretical oversight or inefficient implementation. Consequently, we have endured to fix as many of the problems as possible to see if the obstacle lay in the construction of the algorithm, or in its foundation.

4.2.3 Modified Algorithm

Much of the work done on this project was direct modification to code: rewriting the majority of the subroutines to increase efficiency, and cleaning up unnecessary computations. In particular, the complexities resulting in the transformation between ξ and σ were removed and we transitioned to standardized optimization routines. These changes resulted in the algorithm successful converging in systems with multiple particles.

We begin by noting some of the simple changes made to the theoretical implementation that allowed us to use faster and more general techniques for computation. Recall that we require that the interval times $\xi \in \mathbb{R}^\ell$ be positive. If $\Phi : \mathbb{R}^\ell \to \mathbb{R}$ is an objective function, we are considering the problem

$$\min_{\xi \in \mathbb{R}^\ell} \Phi(\xi) \quad \text{subject to } \xi_i > 0,\ i = 1, \ldots, \ell. \tag{4.21}$$

We may choose to write this in a more concise format as

$$\min_{\xi \in \mathbb{R}^\ell_+} \Phi(\xi), \qquad \mathbb{R}^\ell_+ = \left\{ x = (x_1, \ldots, x_\ell) \in \mathbb{R}^\ell \;\middle|\; x_i > 0, i = 1, \ldots, \ell \right\}. \tag{4.22}$$

Such a restriction forces one to use constrained optimization techniques. However, we can instead cast the system in an equivalent unconstrained framework, which will allow us to use faster and more general optimization algorithms.

Proposition 4.2.1. *Consider the constrained optimization problem given by* (4.22) *and define the function* $\hat{\Phi} : \mathbb{R}^\ell \to \mathbb{R}$ *given by* $\hat{\Phi}(\xi) = \Phi(\xi^2)$*, where we define the notation that* ξ^2 *is taken in a componentwise fashion. More precisely, if* $\Phi(\xi) = \Phi(\xi_1, \ldots, \xi_\ell)$ *then we define* $\hat{\Phi}(\xi^2) = \Phi(\xi_1^2, \ldots, \xi_\ell^2)$*. We claim that the set of minima of* $\hat{\Phi}$ *over all of* $\mathbb{R}^\ell$ *is equivalent to the minima of* Φ *over* $\mathbb{R}^\ell_+$*.*

Proof. Fix an element ξ_i for some $i \in \{1, \ldots, \ell\}$. By taking the partial derivative with respect to ξ_i, we find that

$$\begin{aligned} \frac{\partial}{\partial \xi_i} \hat{\Phi}(\xi) &= \frac{\partial}{\partial \xi_i} \Phi(\xi_1^2, \ldots, \xi_\ell^2) \\ &= 2\xi_i \frac{\partial \Phi}{\partial \xi_i}(\xi_1^2, \ldots, \xi_\ell^2). \end{aligned} \tag{4.23}$$

Let ξ^* be a critical point of Φ. Since each component of ξ^* is non-negative, we can uniquely define $\tilde{\xi} = \left(\sqrt{\xi_1^*}, \ldots, \sqrt{\xi_\ell^*} \right)$ which is in $\mathbb{R}^\ell$. Since ξ^* is critical, we have $\nabla \Phi(\xi^*) = 0$ which implies that that $\frac{\partial \Phi}{\partial \xi_i} \left(\tilde{\xi}^2 \right) = 0$. By (4.23) it follows that $\frac{\partial \hat{\Phi}}{\partial \xi_i}(\tilde{\xi}) = 0$ giving $\nabla \hat{\Phi}(\tilde{\xi}) = 0$ so every critical point of Φ corresponds to a critical point of $\hat{\Phi}$.

Conversely, if ξ^* is a critical point of $\hat{\Phi}$ then $\nabla \hat{\Phi}(\xi^*) = 0$ so $\frac{\partial \hat{\Phi}}{\partial \xi_i} = 0$. By (4.23) we then have that $\xi_i^* \frac{\partial \Phi}{\partial \xi_i} \left(\xi^{*2} \right) = 0$ but since $\xi_i > 0$ this implies $\frac{\partial \Phi}{\partial \xi_i} \left(\xi^{*2} \right) = 0$ and hence $\nabla \Phi \left(\xi^{*2} \right) = 0$ so all critical points of $\hat{\Phi}$ have corresponding critical points of Φ.

The above two implications imply there is a many-to-one surjective correspondence between the critical points of $\hat{\Phi}$ and the critical points of Φ. We now want to show that the minima are preserved. This is a simple matter since the constructions above indicate that the mapping between critical points preserves the value of the function. The unsatisfied reader can note that this follows from the universal property of quotient sets. If $\hat{C}, C$ are the sets of critical points of $\hat{\Phi}$ and Φ respectively, we can define a relation $\sim$ on $\hat{C}$ by $a \sim b$ if $a^2 = b^2$. Then $C \cong \hat{C}/\sim$ and furthermore $a \sim b$ implies that $\hat{\Phi}(a) = \hat{\Phi}(b)$. By the universal property of quotients, if $Z = \hat{\Phi}(\hat{C})$ there exists a unique function $f : C \to Z$ such that $\hat{\Phi} = f \circ \pi$ where $\pi : \hat{C} \to C$ is the canonical quotient projection, $\pi(a) = [a]_\sim = a^2$. Since $\hat{\Phi}(\xi) = \Phi(\pi(\xi))$ and f is unique it follows that $f = \Phi$ and so in fact all extrema agree. □

The most significant computational improvement made to this algorithm was the implementation of a one-time calculation for matrix exponentials. We assume that we are given a prescribed set $(\tilde{H}_1, \ldots, \tilde{H}_\ell)$ of effective Hamiltonians, so that the evolution of the operator X is given by

$$X(\xi) = \prod_{i=0}^{\ell-1} \exp\left[-i\tilde{H}_{\ell-i}\xi_{\ell-i}\right]. \tag{4.24}$$

Since each $\tilde{H}_i$ has a matrix representation, this expression is the composition of many matrix exponentials. Such exponentials are not only very expensive from a computational standpoint, but need to be re-derived each time there is an update to the switching times ξ_i. As we are consistently varying the value of each ξ_i, we would like to avoid needing to recompute the exponential each time we update ξ.

The important thing to notice is that the $\tilde{H}_i$ are fixed and only ξ_i is free to vary. By calculating the eigenvalue decomposition of $\tilde{H}_i$, we can write

$$\tilde{H}_i = UDU^\dagger \tag{4.25}$$

for some diagonal matrix of eigenvalues $D = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ and unitary matrix $U \in \mathfrak{U}(d)$. This decomposition is guaranteed to exist since $\tilde{H}_i \in i\mathfrak{u}(d)$ is Hermitian and hence a normal matrix, allowing us to apply the Spectral Theorem[23].

The reader may notice that the eigenvalue decomposition of a matrix is also a computationally expensive operation. However, we cater to the fact that our algorithm will require more computations of an exponential than the one-time eigenvalue decomposition, and is a promising step to speed up the algorithm [53].

Let $r \in \mathbb{R}$ and notice that

$$\begin{aligned}\exp\left[-ir\tilde{H}_i\right] &= \exp\left[-irUDU^\dagger\right] \\ &= U\exp\left[-irD\right]U^\dagger \\ &= U\left(\exp\left[-iD\right]\right)^r U^\dagger.\end{aligned}$$

Since D is diagonal it follows that $\exp[-iD] = \mathrm{diag}(e^{-i\lambda_1}, \ldots, e^{-i\lambda_d})$ is also diagonal, and more importantly

$$(\exp[-iD])^r = \mathrm{diag}(e^{-ir\lambda_1}, \ldots, e^{-ir\lambda_d}). \tag{4.26}$$

With a single eigendecomposition of each $\tilde{H}_i$, we can easily compute the matrix exponential $\exp\left[-ir\tilde{H}_i\right]$ for any time $r \in \mathbb{R}$. This allows us to quickly derive the forward propagator $X(\xi)$ for any vector of switching times ξ.

We note however that the unitary matrix U and the eigenvalues given in D need to be stored for *each* effective Hamiltonian. This results in an exponential increase in the amount of memory required, though this is often a preferred exchange in favour of decreased run-times.

The efficiency of this exponential calculation is available because of the assumption that the controls are bang-bang. While the act of calculating the matrix exponential does not scale well as the size of the system increases, the fact that we need only compute it once for each effective Hamiltonian could provide computational benefits. Nonetheless, it seems as though there are some key theoretical reasons why the algorithm in its current implementation is intractable. The following sections will preview some of the work done to recast the problem into a more amenable framework.

4.3 Attempt to Embed the Problem in $\mathbb{R}^\ell$

With the the efficiency increase and changes made to the Kaya-Huneault algorithm above, we were able to see excellent performance and convergence for as many as three-qubits. However, as we extend to more dimensions the implementation times given by the solutions became very large, suggesting that they might not be optimal. The issue with slow run-times can attributed to doing much of our work with matrix-valued functions which result in large dimensional calculations of the gradient for even a small number of states. We hypothesized that if we could instead embed the problem into real space, not only could we apply classical numerical methods but we might also be able to dramatically reduce the dimension of the problem.

4.3.1 The Real-Embedding Algorithm

Let $X_d \in \mathfrak{U}(d)$ describe our desired unitary transformation. Recall that our objective function that measured distance was given by $f(X) = d(X, X_d)^2$ with $d(\cdot,\cdot)$ the metric induced under the Hilbert-Schmidt inner product. We defined a forward propagation operator from (4.9) as

$$g_\ell : \mathbb{R}_+^\ell \times (i\mathfrak{u}(d))^\ell \to \mathfrak{U}(d), \qquad \left(\xi, (\tilde{H}_1, \ldots, \tilde{H}_\ell)\right) \mapsto \prod_{i=0}^{\ell-1} \exp\left[-i\tilde{H}_{\ell-i}\xi_{\ell-i}\right].$$

For a fixed, prescribed set of effective Hamiltonians $(\tilde{H}_1, \ldots, \tilde{H}_\ell)$ we can define the function

$$g : \mathbb{R}^\ell \to \mathfrak{U}(d), \qquad \xi \mapsto g_\ell\left(\xi, (\tilde{H}_1, \ldots, \tilde{H}_\ell)\right). \tag{4.27}$$

By composing the functions f and g we are able to consider a mapping between real spaces as

$$F = f \circ g : \mathbb{R}^\ell \to \mathbb{R}, \qquad F(\xi) = d\left(\prod_{i=0}^{\ell-1} \exp\left[-i\tilde{H}_{\ell-i}\xi_{\ell-i}\right] X_0, X_d\right) \tag{4.28}$$

and from here we can apply standard Euclidean optimization techniques.

Our goal will be to create a numerical algorithm that is able to minimize the function $F(\xi)$ while taking into consideration the time of the synthesis. In order to define the function F as a purely real objective function, there was an implicit assumption about the appropriate value for $\ell \in \mathbb{N}$ and the selection of a prescribed collection of $\tilde{H}_i$. Once these quantities are assumed, we can then focus on discovering the intervals ξ_i that are optimal to the unitary evolution. Unfortunately, the theoretical considerations thus far do not impart any information about the structure of the optimal choice of effective Hamiltonians, nor even how many switching intervals there should be. Properly deducing a time-optimal solution necessitates considering all three conditions simultaneously. Since it is intractable to consider such a simultaneous approach, we instead embrace the technique often employed in numerics: the estimation of two of the parameters followed by a computational solution of the remaining unknown parameter.

We attempt a numerical algorithm that involves setting a fixed value for $\ell \in \mathbb{N}$ and guessing an initial set of controls $\tilde{H}_i$. With these approximations, our problem is reduced to one that looks like a standard constrained optimization problem:

$$\min_{\xi \in \mathbb{R}^\ell} \sum_{i=1}^{\ell} \xi_i \qquad \text{subject to } F(\xi) = 0 \tag{4.29}$$

where $F(\xi)$ is given in (4.28).

We shall use the same generic technique as described in section 4.2.1. Define a terminal surface $\Omega = F^{-1}(0) \subseteq \mathbb{R}^\ell$ and attempt to find a point $\xi_0 \in \Omega$. After such a point is found, attempt to minimize $\sum_{i=1}^{\ell} \xi_\ell$ via a continuous optimization method. This algorithm will be known as the *Real-Embedding algorithm* and the pseudo-code is given below.

Let $X_d \in \mathfrak{U}(d)$ be the objective unitary transformation. Let $\ell \in \mathbb{N}$ describe the number of discretized intervals, and prescribe a set $(\tilde{H}_1, \ldots, \tilde{H}_\ell)$ of time-independent effective Hamiltonians on each interval. Define $F(\xi)$ as in (4.28) and proceed as follows:

1. Find a point on the terminal surface $\xi_0 \in F^{-1}(0)$. Equivalently, calculate an initial set of switching intervals ξ_0 such that $X(\xi_0) = X_d$.
2. Using ξ_0 as an initial point, solve the constrained optimization problem (4.29). Let this point be $\hat{\xi}$.
3. If $\hat{\xi}$ does not satisfy the constraints, a maximum iteration counter has been exceeded, or the change in $\hat{\xi}$ between iterations satisfies a lower bound, terminate the algorithm. Remove collapsed intervals and merge similar intervals as necessary.
4. Set $\xi_0 = \hat{\xi}$ and append zero-length intervals to the beginning and end of the control sequence. Go to step 2.

We note that there may be topological problems shared by both the Real-Embedding and Kaya-Huneault methods. We refer to Section 4.3.3 for more information.

This algorithm offers many improvements over the modified Kaya-Huneault algorithm. The Real-Embedding problem converges more often and with lower run-times than Kaya-Huneault, although at the cost of occasionally finding poor synthesis times. We will see in Section 4.4 that the only case when the Kaya-Huneault algorithm truly outperforms the Real-Embedding problem is in the single-particle case. Further, the theoretical description of the problem is greatly simplified, with the complexity arising due to quantum mechanics being hidden in the expression of the function $F(\xi)$. We will consider a numerical comparison of this algorithm with Kaya-Huneault and GRAPE in the Section 4.4, but we first wish to elaborate on an important but subtle feature of this technique that may result in issues as we explore a greater number of particles and generalizations. This issue stems from the general theory of constrained optimization via Lagrange multipliers.

4.3.2 Problem with Constrained Optimization

In order to consider why the Real-Embedding problem might run into trouble, we need to introduce a few definitions fundamental to the study of constrained optimization.

Definition 4.3.1 (Nocedal [60])**.** A *real, constrained, optimization problem* is any problem formatted as follows:

$$\min_{x\in\mathbb{R}^n} s(x) \qquad \text{subject to} \qquad \begin{cases} c_i(x) = 0 & i \in \mathcal{E} \\ c_i(x) \geq 0 & i \in \mathcal{I} \end{cases} \tag{4.30}$$

where $\mathcal{E}, I \subset \mathbb{N}$ are index sets. We say that $\mathcal{E}$ is the index set of *equality constraints* and that $\mathcal{I}$ is the index set of *inequality constraints*.

Definition 4.3.2. Given the constrained optimization problem from Definition 4.3.1, we define the *feasible set* as

$$\Omega = \left\{ x \in \mathbb{R}^n \;\middle|\; c_i(x) = 0 \text{ and } c_j(x) \geq 0, i \in \mathcal{E}, j \in \mathcal{I} \right\} \tag{4.31}$$

so that Ω is the subset of $\mathbb{R}^n$ on which the optimization is to occur. Furthermore, given a point $x \in \Omega$ we define the *active set of* x as $\mathcal{A}(x)$ given by

$$\mathcal{A}(x) = \mathcal{E} \cup \left\{ i \in \mathcal{I} \;\middle|\; c_i(x) = 0 \right\}. \tag{4.32}$$

The active set at x consists of all those constraints that are identically zero at x. This motivates the statement that a constraint c_i is *active at* x if $c_i(x) = 0$. All equality constraints are inherently active.

Definition 4.3.3. Consider the constrained optimization problem (4.30). We say that $\hat{x}$ is a *local solution* if there exists a neighbourhood $\mathcal{N}$ of $\hat{x}$ such that $s(\hat{x}) \leq s(x)$ for all $x \in \mathcal{N} \cap \Omega$.

The following constraint qualification is the assumption most used in the design of constrained optimization algorithms. We will see how crucially important it is in providing first-order optimality conditions.

Definition 4.3.4 (Linear Independence Constraint Qualification, Nocedal [60])**.** Consider the constrained optimization problem given in Definition 4.3.1. Given a point $x \in \Omega$ and its active set $\mathcal{A}(x)$, we say that the *Linear Independence Constraint Qualification*(LICQ) holds at x if the set of active constraint gradients $\{\nabla c_i(x), i \in \mathcal{A}(x)\}$ is linearly independent.

This turns out to be a very important assumption in the proof of the following theorem which describes the first order necessary conditions for optimality.

Theorem 4.3.5 (Nocedal [60])**.** *Consider the constrained optimization problem given in* (4.30) *and assume that f and c_i are continuously differentiable functions $i \in \mathcal{I} \cup \mathcal{E}$. Let Ω be the feasible set and $\hat{x}$ be a local solution to* (4.30)*. If the LICQ holds at $\hat{x}$, there exists a Lagrange multiplier $\hat{\lambda}$ with components $\hat{\lambda}_i, i \in \mathcal{E} \cup \mathcal{I}$ such that if we define the Lagrangian function*

$$\mathcal{L}(x, \lambda) = s(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x) \tag{4.33}$$

then the following conditions are satisfied at $(\hat{x}, \hat{\lambda})$

$$\nabla_x \mathcal{L}(\hat{x}, \hat{\lambda}) = 0, \tag{4.34a}$$
$$c_i(\hat{x}) = 0, \qquad \forall i \in \mathcal{E}, \tag{4.34b}$$
$$c_i(\hat{x}) \geq 0, \qquad \forall i \in \mathcal{I}, \tag{4.34c}$$
$$\hat{\lambda}_i \geq 0, \qquad \forall i \in \mathcal{I}, \tag{4.34d}$$
$$\hat{\lambda}_i c_i(\hat{x}) = 0, \qquad \forall i \in \mathcal{E} \cup \mathcal{I} \tag{4.34e}$$

The conditions given by (4.34) *are known as the* Karush-Kuhn-Tucker*(KKT) conditions.*

If we now return to our problem (4.29), we can easily convert the Real-Embedding problem into the generalized form by setting $\mathcal{E} = \{1\}, \mathcal{I} = \emptyset$ and

$$s(\xi) = \sum_{i=1}^{\ell} \xi_i, \qquad c_1(\xi) = F(\xi). \tag{4.35}$$

Let us examine what happens if try to apply the KKT conditions to (4.35). Our definition of the terminal surface $\Omega = F^{-1}(0)$ corresponds to the feasible set given in Definition 4.3.2 and so there is no ambiguity of notation. We notice that

$$F(\xi) = d(g(\xi)X_0, X_d)^2 \tag{4.36}$$

and hence $F(\xi) \geq 0$ since it is the square of a function. Even if one removes the square, the function is still non-negative by definition of the metric. Furthermore, F is a differentiable function since it is the composition of differentiable functions, so that any point ξ_0 at which $F(\xi_0) = 0$ must correspond to a minimum of F. Interior minima of differentiable functions are necessarily critical points, so

$$\nabla F(\xi) = 0, \qquad \forall \xi \in \Omega. \tag{4.37}$$

With our Lagrangian function given by $L(x, \lambda) = s(\xi) - \lambda F(\xi)$, we see that since $s(\xi)$ is just a sum of the components of ξ then $\nabla_\xi s(\xi)$ will just be a vector of ones, which we will denote by $\vec{1}$. If we apply the gradient to the Lagrangian we find that

$$\begin{aligned} \nabla_\xi \mathcal{L}(\xi, \lambda) &= \nabla_\xi s(\xi) - \lambda \nabla_\xi F(\xi) \\ &= \vec{1} + \lambda \nabla_\xi F(\xi). \end{aligned} \tag{4.38}$$

If $\hat{\xi} \in \Omega$ is a local solution to (4.29), we can apply (4.34a) of Theorem 4.3.5 to find that (4.38) becomes

$$\nabla_\xi \mathcal{L}(\hat{\xi}, \lambda) = \vec{1} + \lambda \nabla_\xi F(\hat{\xi}) = \vec{1} \tag{4.39}$$

since $\nabla_\xi F(\hat{\xi}) = 0$ by (4.37). On the other hand, (4.34a) implies that $\nabla_\xi \mathcal{L}(\hat{\xi}, \lambda) = 0$ which yields a contradiction. Since $\nabla F(\xi) = 0$ for *every* feasible point, (4.39) certainly holds when ξ is replaced by the optimal solutions $\hat{\xi}$. This occurs because the zero-vector is vacuously linearly dependent so the LICQ is violated and we cannot apply Theorem 4.3.5.

An alternative approach may be to change the regularity conditions on solutions to (4.30), though in most instances one will still be unable to apply the KKT conditions. This occurs because most constraint qualifications require some level of linear independence of the active constraint set [60]. Furthermore, using Lagrange multipliers is intractable as indicated by the contradiction arrived at in (4.39).

4.3.3 Topological Issues

We have discussed the Kaya-Huneault algorithm in Section 4.2.1 and its shortcomings in Section 4.2.2. This was followed by the Real-Embedding algorithm in Section 4.3.1 and its issues in Section 4.3.2. There is another obstacle that is shared by both of these methods, which we discuss here.

A possible caveat occurs when one tries to examine the terminal landscape itself. We have very little information about the basic topological structure of this subspace, such as whether it is connected, open, or even has non-zero measure. Since the feasible set used in our constrained optimization method is defined by the minima of the distance measure function, we can analyze this function and hope to gain information about its structure.

Continuous optimization results require the feasible set to be continuous. For example, simplex and interior point methods move in a continuous fashion on the boundary and interior of the feasible set, respectively. Such algorithms will break down if the feasible set

contains only isolated points. We suspect that this may be one of the problems that our algorithm is facing.

We heuristically corroborate this hypothesis by computational simulation in the instance of bounded controls. Figure 4.1 gives a small section of the distance measure as it varies with respect to two switching intervals. We notice that it is highly oscillatory and contains multiple minima. Figure 4.2 shows several level sets of Figure 4.1, representing the points where the distance between the evolved state and the objective function is equal to $\{0.01, 0.05, 0.1, 0.2\}$. In the one particle case, this corresponds to fidelities of $\{99.9975\%, 99.9375\%, 99.75\%, 99.0\%\}$.

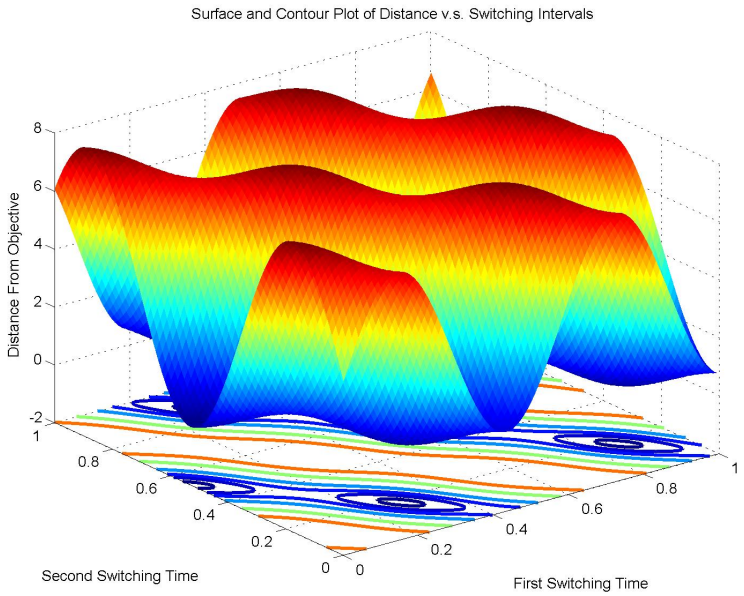

Figure 4.1: A surface and contour plot of the distance measure as a function of switching intervals.

Using the information from Figure 4.2 and the fact that, as we decrease the value of the level sets the area of these regions decreases, it seems likely that the terminal surface may consist entirely of isolated points. In such a case, the continuous optimization subroutines employed by the Kaya-Huneault and Real-Embedding algorithms would fail. Figure 4.3 is also included to support this hypothesis, and plots the one-dimensional cross sections of a two-level, two-particle scenario.

If the feasible set can be shown to contain only isolated points, then the continued use

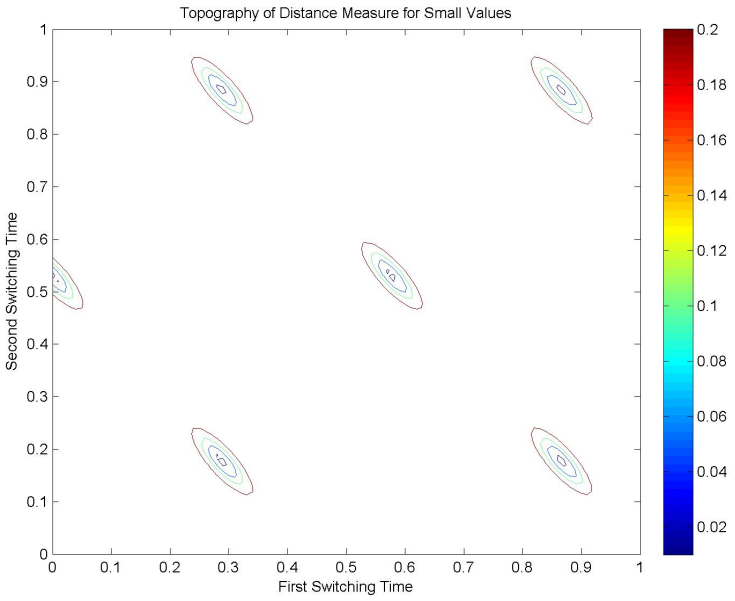

Figure 4.2: A graph of level sets corresponding to Figure 4.1 for small values.

of the ideas presented in these algorithms necessitates using combinatorial optimization. This may still be intractable, as combinatorial optimization problems often assume that the problem is presented in a manner in which we are given or can deduce the position of all points in the feasible set. We do not have this ability for this problem.

Finally, finding an initial point on the terminal surface may also prove to be very difficult, and this step is often required to initialize constrained optimization routines. Simulations reveal that points of perfect control correspond to sharp peaks, resulting in very steep gradients and small neighbourhoods of descent near points on the terminal surface. This is again supported by Figure 4.3.

4.4 Numerical Comparison

We are now ready to begin numerical simulation to test the suitability of these algorithms in different environments. We begin by introducing the form of the problem, as well as the individual systems themselves and a set of objective gates. The gates we will be considering are all important in the theory of quantum information[9, 29, 44, 57]. This will be followed

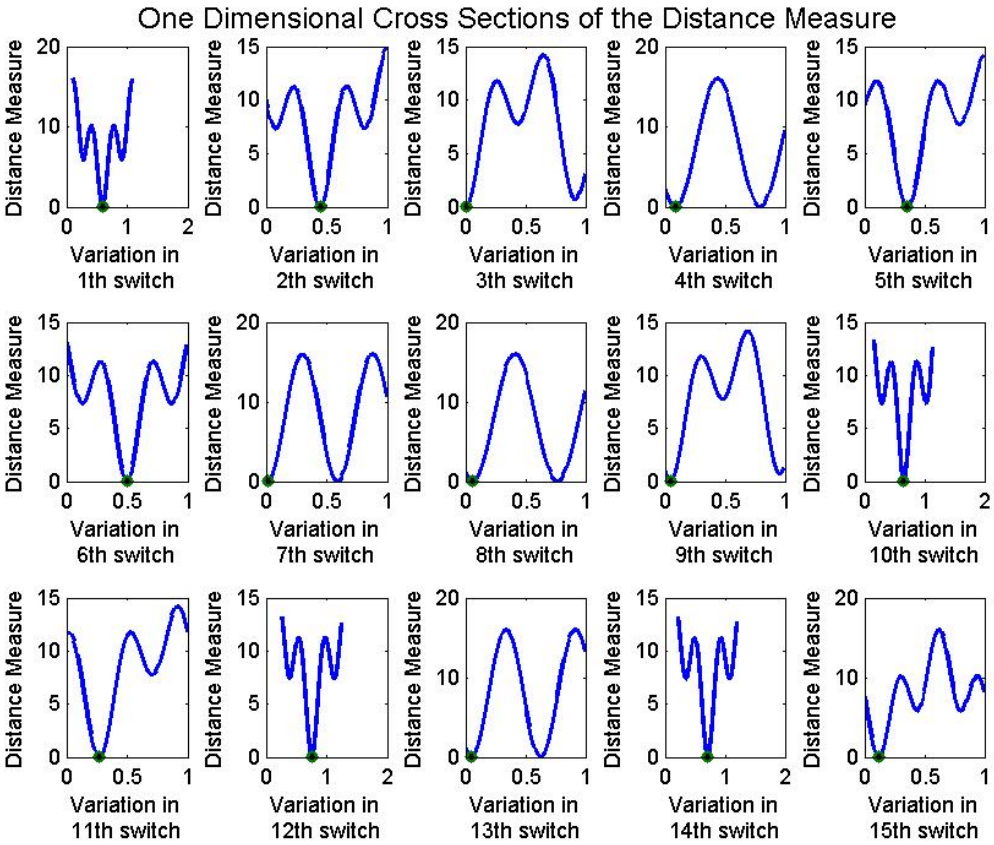

Figure 4.3: The one dimensional cross sections near a minimum of the distance measure in a two qubit system.

by a brief discussion on the comparison of the methods for each system.

4.4.1 Numerical Setup

We begin by establishing the environment in which the simulations are done. We are looking at evolution as dictated by the Schrödinger equation

$$\frac{d}{dt}U(t) = -iH(u(t))U(t), \qquad U(t_0) = U_0 \tag{4.40}$$

where from this point on we will assume that $t_0 = 0$. We will test the algorithms on very simple arrangements varying from single-particle systems to three-particle systems. In each of these three cases, three different objective unitary operators will be tested. As we have done before, assume that the Hamiltonian is determined by an affine, m-input form described by

$$H(u(t)) = H_0 + \sum_{i=1}^{m} u_i(t)H_i \tag{4.41}$$

with normalized control amplitudes $u_i(t) \in [0, 1]$. As it will be useful in concisely defining our Hamiltonians, we introduce here the Pauli matrices given by

$$X = \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Y = \frac{1}{2}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad Z = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{4.42}$$

By taking the 2-dimensional identity matrix I_2, it is shown in Appendix C.1 that $\{I, X, Y, Z\}$ is a basis for $\mathfrak{u}(2)$ and $\{X, Y, Z\}$ is a basis for $\mathfrak{su}(2)$. We can extend this basis to $\mathfrak{u}(2^n)$ and $\mathfrak{su}(2^n)$ by considering the Kronecker product of these basis elements, and will be using these as the fundamental building blocks for most of our systems.

The GRAPE algorithm we are using has been modified from its original purpose to serve as a time-optimal implementation. In particular, we have adapted GRAPE to take a set of initial parameters describing the number of discretizations N of the time interval, as well as the uniform size of each sub-interval Δt. By specifying these conditions, we see that the overall time is fixed and given by $N\Delta t$ with partition

$$0 < \Delta t < 2\Delta t < \ldots < (N-1)\Delta t < N\Delta t. \tag{4.43}$$

The *time-optimal* GRAPE algorithm then performs a bisection algorithm to reduce the

total synthesis time. The pseudo-code is given as follows:

Given an initial specification of the discretization number N, a uniform sub-interval length Δt, a fidelity lower bound ϵ, and an objective unitary X_d, proceed as follows:

1. Initialize $k = 1, N_0 = N$ and apply GRAPE to find a pulse sequence that drives the identity to X_d over time $N_0 \Delta t$.
2. If GRAPE does not find a solution satisfying the fidelity tolerance ϵ, STOP. Otherwise, define $N_1 = N_0 - \frac{N}{2}$.
3. Apply GRAPE to find a pulse sequence over time $N_k \Delta t$.
4. If GRAPE finds a solution satisfying the fidelity ϵ, set $N_{k+1} = N_k - \frac{N}{2^{k+1}}$. Otherwise, set $N_{k+1} = N_{k-1} + \frac{N}{2^{k+1}}$ and update $k = k + 1$.
5. Check stopping criterion. Namely, if $N_k \Delta t$ is less that some tolerance or k has exceeded the maximum number of halving, STOP.
6. GOTO 3.

In order to read the tables, we define the shorthand notations: "GR" for GRAPE, "KH" for the Kaya-Huneault algorithm, and "RE" for the algorithm used in solving the Real-Embedding problem. We have used ∞ to denote instances in which the algorithm did not converge and signifies that the data entry is not applicable as there is no information to display. Furthermore, all tables include "CPU-time" and "Run Time" sections. The CPU-time represents the amount of time each processor used when at full capacity. In such a setting, 1 second of real-time at 50% capacity represents 0.5 CPU-seconds while 1 second of real-time with four processors at 100% capacity represents 4 CPU-seconds. Conversely, the run time category represents the real time of the computation. The purpose of giving both numbers is to demonstrate the parallel computing abilities of each algorithm giving real-time expectations for results, but at the same time give an overall example of the total computational resources necessary to compute the problem. The KH and RE columns represent an average over multiple runs, and so the reader should regard this information with caution.

We note that GRAPE uses fidelity measure while KH and RE use distance measures. If the distance measure is given by m_{dis} and the fidelity is given by m_{fid}, the conversion between the two for a system occupying $\mathfrak{U}(2^d)$ is given by the following equations:

$$m_{\text{fid}} = 1 - \frac{m_{\text{dis}}^2}{2^{d+1}}, \qquad m_{\text{dis}} = \sqrt{2^{d+1}\left(1 - m_{\text{fid}}\right)} \tag{4.44}$$

which can be derived from the calculations in section 4.2.1 concerning equation (4.3).

All simulations were coded in Matlab R2009b and run on a Sun x4600 server including eight Opteron 8218 processors with 32 GB of RAM.

4.4.2 One-Particle System

Herein we consider a two-input system, $m = 2$, with drift Hamiltonian

$$H_0 = 2\pi \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = 4\pi Z \tag{4.45}$$

and control Hamiltonians

$$H_1 = 2\pi \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = 4\pi X$$
$$H_2 = 2\pi \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = 4\pi Y.$$

The three objective special unitaries that we will be considering are as follows:

$$X_{\text{Had}} = \frac{i}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad X_{\text{NOT}} = i \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad X_{\frac{\pi}{2}} = \frac{1-i}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \tag{4.46}$$

all of which have important implications in the theory of quantum computation. The convergence threshold was set at a fidelity of 99.95%.

Consider Table 4.1 as it relates to the operator $X_{\text{Had},2}$. The GRAPE algorithm took significantly longer than KH or RE, but yielded much better results in terms of synthesis time. Also, an appropriate choice of parameters for GRAPE made a dramatic difference in run time, and KH and RE both found consistent solutions in terms of synthesis time. This may imply that KH and RE found the time optimal solution as permitted by bang-bang controls and that they were not able to access the solution afforded to GRAPE. Further, KH was able to find these solutions very quickly compared to RE.

For X_{NOT} we consider Table 4.2 and notice precisely the same pattern as $X_{\text{Had},2}$. GRAPE provided the best synthesis times at the cost of being much slower than KH and RE, and once again the appropriate choice of initial parameters caused GRAPE to converge more than $4\times$ faster than the original conditions. In this case, KH and RE found similar results in a similar time-frame and so we judge that they were equal.

Table 4.1: One-Particle Simulation: $X_{\mathrm{Had},2}$

Initial Parameters	Property	Algorithm GR	KH	RE
GRAPE Steps = 1000 Length = 5×10^{-4} KH/RE Intervals = 2	CPU Time (secs)	354.9724	0.7565	5.8751
	Run Time (secs)	356.1582	0.7615	6.1410
	Syn. Time ($\hbar$-secs)	0.1745	0.4648	0.4655
GRAPE Steps = 1000 Length = 5×10^{-3} KH/RE Intervals = 3	CPU Time (secs)	132.0911	0.2045	11.6384
	Run Time (secs)	131.9476	0.2098	11.7638
	Syn. Time ($\hbar$-secs)	0.1750	0.5250	0.5303
GRAPE Steps = 1000 Length = 1×10^{-3} KH/RE Intervals = 5	CPU Time (secs)	22.8217	2.8965	23.5641
	Run Time (secs)	22.7722	2.9000	23.6642
	Syn. Time ($\hbar$-secs)	0.1750	0.4645	0.4655

Table 4.2: One-Particle Simulation: X_{NOT}

Initial Parameters	Property	Algorithm GR	KH	RE
GRAPE Steps = 1000 Length = 5×10^{-4} KH/RE Intervals = 2	CPU Time (secs)	359.7724	1.4585	1.0928
	Run Time (secs)	360.7575	1.4617	1.2139
	Syn. Time ($\hbar$-secs)	0.2915	∞	∞
GRAPE Steps = 1000 Length = 1×10^{-3} KH/RE Intervals = 3	CPU Time (secs)	175.5832	2.3845	1.1459
	Run Time (secs)	175.5410	0.1522	1.1596
	Syn. Time ($\hbar$-secs)	0.1910	∞	∞
GRAPE Steps = 1000 Length = 5×10^{-3} KH/RE Intervals = 5	CPU Time (secs)	40.9474	14.3475	15.1635
	Run Time (secs)	40.9024	14.3475	15.3722
	Syn. Time ($\hbar$-secs)	0.1950	0.4499	0.5308

Table 4.3: One-Particle Simulation: X_{NOT}

Initial Parameters	Property	Algorithm GR	KH	RE
GRAPE Steps = 1000 Length = 5×10^{-4} KH/RE Intervals = 2	CPU Time (secs)	1577.8026	0.8800	7.4751
	Run Time (secs)	1580.4226	0.8836	7.4942
	Syn. Time ($\hbar$-secs)	0.2915	0.6579	0.6611
GRAPE Steps = 1000 Length = 1×10^{-3} KH/RE Intervals = 3	CPU Time (secs)	584.4762	0.3295	29.4090
	Run Time (secs)	615.7718	0.3300	20.4772
	Syn. Time ($\hbar$-secs)	0.2920	0.6580	0.1250
GRAPE Steps = 1000 Length = 5×10^{-3} KH/RE Intervals = 5	CPU Time (secs)	18.8951	5.6020	12.1782
	Run Time (secs)	19.5961	5.6081	12.2044
	Syn. Time ($\hbar$-secs)	0.1350	0.1247	0.1250

Finally, Table 4.3 gives us the information for $X_{\frac{\pi}{2}}$. Here we notice a different trend. Similar to our previous gates, the first set of initial conditions resulted in a slow running GRAPE providing the best solution. However, the second set of conditions resulted in KH and RE providing better overall optimal solutions, despite GRAPE running quickly. Here KH is clearly the best algorithm, as it was able to find the best solution and do so in an expedient manner.

4.4.3 Two-Particle System

We move on to a four-input system, $m = 4$, with drift Hamiltonian

$$H_0 = 2\pi Z \otimes Z \tag{4.47}$$

and control Hamiltonians

$$\begin{aligned} H_1 &= 2\pi X \otimes I_2 & \qquad H_3 &= 2\pi I_2 \otimes X \\ H_2 &= 2\pi Y \otimes I_2 & \qquad H_4 &= 2\pi I_2 \otimes Y. \end{aligned}$$

The three objective unitaries that we will be considering are as follows:

$$X_{\text{CNOT}} = \frac{1+i}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \qquad X_{\text{Had,4}} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix},$$

$$X_{\text{SWAP}} = \frac{1+i}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

The convergence threshold was set at a fidelity of 99.98%.

Table 4.4, Table 4.5 and Table 4.6 all exhibit similar properties, allowing us to discuss them in general. We first notice that GRAPE has long run-times ranging from 2 to 4 times the length of KH and 2 to 20 times as long as RE. As in the single-particle case, GRAPE still provides the best synthesis time, though RE stays competitive providing solutions that are approximately twice as long. KH in this instance is starting to fail, as its run time exceeds RE and its solutions are significantly worse than either GRAPE or RE.

Table 4.4: Two-Particle Simulation: X_{CNOT}

Initial Parameters	Property	Algorithm GR	KH	RE
GRAPE Steps = 1000 Length = 5×10^{-3} KH/RE Intervals = 20	CPU Time (secs)	2069.60	521.05	147.42
	Run Time (secs)	2068.91	521.60	40.85
	Syn. Time ($\hbar$-secs)	1.1500	21.1000	2.8466
GRAPE Steps = 2000 Length = 2.5×10^{-3} KH/RE Intervals = 30	CPU Time (secs)	6424.23	400.47	292.95
	Run Time (secs)	6425.29	400.47	66.64
	Syn. Time ($\hbar$-secs)	0.9350	14.7100	2.6873

Table 4.5: Two-Particle Simulation: $X_{\text{Had},4}$

Initial Parameters	Property	Algorithm GR	KH	RE
GRAPE Steps = 1000 Length = 5×10^{-3} KH/RE Intervals = 20	CPU Time (secs)	1018.92	461.63	114.34
	Run Time (secs)	1024.99	462.02	36.44
	Syn. Time ($\hbar$-secs)	1.4900	31.1050	3.1940
GRAPE Steps = 2000 Length = 2.5×10^{-3} KH/RE Intervals = 30	CPU Time (secs)	764.15	564.79	355.82
	Run Time (secs)	757.09	564.88	73.97
	Syn. Time ($\hbar$-secs)	1.4850	19.1200	3.1083

Table 4.6: Two-Particle Simulation: X_{SWAP}

Initial Parameters	Property	Algorithm GR	KH	RE
GRAPE Steps = 1000 Length = 5×10^{-3} KH/RE Intervals = 20	CPU Time (secs)	1063.22	530.26	51.217
	Run Time (secs)	1073.68	530.57	27.28
	Syn. Time ($\hbar$-secs)	1.4900	31.32	2.1480
GRAPE Steps = 2000 Length = 2.5×10^{-3} KH/RE Intervals = 30	CPU Time (secs)	764.15	377.36	360.413
	Run Time (secs)	757.09	377.47	74.95
	Syn. Time ($\hbar$-secs)	1.4850	12.97	2.4198

4.4.4 Three-Particle System

Finally, we consider a six-input system $m = 6$, with drift Hamiltonian

$$H_0 = 2\pi(Z \otimes Z \otimes I_2 + I_2 \otimes Z \otimes Z) \tag{4.48}$$

and control Hamiltonians

$$\begin{aligned} H_1 &= 2\pi X \otimes I_2 \otimes I_2 & H_4 &= 2\pi I_2 \otimes Y \otimes I_2 \\ H_2 &= 2\pi Y \otimes I_2 \otimes I_2 & H_5 &= 2\pi I_2 \otimes I_2 \otimes X \\ H_3 &= 2\pi I_2 \otimes X \otimes I_2 & H_6 &= 2\pi I_2 \otimes I_2 \otimes Y. \end{aligned}$$

The three objective unitaries that we will be considering are as follows:

$$X_{\text{Fred}} = C \left(\begin{array}{c|cc|c} I_5 & \multicolumn{2}{c|}{0} & 0 \\ \hline & 0 & 1 & \\ 0 & 1 & 0 & 0 \\ \hline 0 & \multicolumn{2}{c|}{0} & 1 \end{array}\right), \qquad X_{\text{Toff}} = C \left(\begin{array}{c|cc} I_6 & \multicolumn{2}{c}{0} \\ \hline & 0 & 1 \\ 0 & 1 & 0 \end{array}\right)$$

$$X_{\text{Had},8} = \frac{1}{2} X_{\text{Had},4} \otimes X_{\text{Had},4}$$

where $C = \frac{1}{2}\left(\sqrt{2+\sqrt{2}} - \sqrt{2-\sqrt{2}}\right)$ is a normalization constant. The convergence threshold was set at a fidelity of 99.99%.

Table 4.7: Three-Particle Simulation: X_{Fred}

Initial Parameters	Property	Algorithm GR	KH	RE
GRAPE Steps = 1000 Length = 5×10^{-3} KH/RE Intervals = 70	CPU Time (secs)	2457.02	663.56	506.13
	Run Time (secs)	2486.15	680.34	69.57
	Syn. Time ($\hbar$-secs)	2.9750	134.5566	43.1727
GRAPE Steps = 2000 Length = 2×10^{-3} KH/RE Intervals = 80	CPU Time (secs)	4705.32	1395.68	680.04
	Run Time (secs)	4681.20	1425.72	69.57
	Syn. Time ($\hbar$-secs)	2.3980	150.9300	56.9611

Let us examine Tables 4.7, 4.8 and 4.9. It can be seen that KH, while maintaining a competitive run-time with RE, finds solutions that are on the order of 10^2 of those given by GRAPE. RE on the other hand is giving solutions on the order of 10^1 longer compared to GRAPE, and so again are not even approximate time-optimal solutions. GRAPE is still yielding excellent times, albeit once again at extreme time cost. Compared to RE, GRAPE is consistently on the order of 10^1 times longer in CPU-time and 10^2 in run-time. There is a version of GRAPE which is made for parallel computing which would decrease its run-time. However, Matlab R2009b does not natively support parallel processing for any of the sub-routines used in our version of the code. We advise the reader to use the CPU-time to determine the speed of such algorithms.

It can now be seen that RE is starting to have trouble finding candidate time-optimal solutions and that KH is even further away than in the two-particle system. GRAPE is still finding good solutions though the amount of computational resources remains high.

4.4.5 Discussion

The one-particle case proved to be interesting as there was a gate in which the bang-bang algorithms provided the best control sequence. However, once the number of particles started to increase, KH and RE were simply unable to find solutions that could compete with GRAPE. However, the downside to a time-optimal GRAPE implementation arises

Table 4.8: Three-Particle Simulation: X_{Had}

Initial Parameters	Property	Algorithm GR	KH	RE
GRAPE Steps = 1000 Length = 5×10^{-3} KH/RE Intervals = 70	CPU Time (secs)	4760.61	663.56	468.47
	Run Time (secs)	4831.66	680.34	50.56
	Syn. Time ($\hbar$-secs)	1.8600	134.5566	32.4540
GRAPE Steps = 2000 Length = 2×10^{-3} KH/RE Intervals = 80	CPU Time (secs)	9689.31	1395.68	722.77
	Run Time (secs)	9623.31	1425.72	73.09
	Syn. Time ($\hbar$-secs)	1.2360	150.9348	46.0556

Table 4.9: Three-Particle Simulation: X_{Toff}

Initial Parameters	Property	Algorithm GR	KH	RE
GRAPE Steps = 1000 Length = 5×10^{-3} KH/RE Intervals = 70	CPU Time (secs)	5158.90	673.32	509.134
	Run Time (secs)	5282.34	682.41	54.81
	Syn. Time ($\hbar$-secs)	2.4250	136.5030	35.4780
GRAPE Steps = 2000 Length = 2×10^{-3} KH/RE Intervals = 80	CPU Time (secs)	4293.66	1134.85	707.67
	Run Time (secs)	4273.66	1135.78	74.24
	Syn. Time ($\hbar$-secs)	2.4640	150.7099	54.7661

in the amount of time it takes to compute solutions, which the tables indicate climbs dramatically as the size of the system increases.

The inability of the KH and RE algorithms to find solutions poses an interesting question. The KH and RE algorithms assume that time optimal solutions are bang-bang and neglect the existence of singular controls: the GRAPE algorithm makes no such assumption. It is possible that GRAPE's ability to find better implementation times could be a theoretical result rather than just a limitation of the numerical methods used to implement KH and RE. In particular, GRAPE may have access to solutions that bang-bang algorithms simply cannot find.

Indeed, we posit that this may be the case. We recall from Section 3.2.3 that [16] analyzed time-optimal controls for a single particle in a magnetic field. The authors were able to relate the control laws discovered for state-transition to one of propagator evolution by using the diffeomorphism between S^3 and $\mathfrak{SU}(2)$. Certain time-optimal solutions required the use of singular controls. It seems reasonable that this may also be the case as more particles are used. This would mean that bang-bang algorithms such as the Real-Embedding algorithm and the Kaya-Huneault algorithm are not well suited to computing time-optimal solutions.

Our results suggest that using bang-bang algorithms with real parameters may not be the best method of approach. We recall that the Real-Embedding algorithm in particular removed the geometric and structural information of the unitary group by considering the real-valued cost of a real-parameterized curve on $\mathfrak{U}(d)$. Instead of ignoring the intermediate details, we may be able to exploit properties of the unitary group to facilitate the creation of better algorithms. With this in mind, we have done some work with geometric algorithms whose results can be found in Appendix D.

The aforementioned geometric algorithms are a combination of the techniques and theories used in the Real-Embedding and GRAPE methods. GRAPE assumes that the pulse function is continuous and discretizes into piecewise constant intervals for numerical reasons. Conversely, the Real-Embedding algorithm assumes the control is bang-bang causing the evolution to be piecewise constant by construction. The geometric algorithms that we have analyzed do not make the assumption that the control is either bang-bang or continuous. Instead, these algorithms use piecewise constant fields but do not limit the controls to occupy vertices of the admissible set as done in bang-bang theories. This will result in fewer per-iteration calculations as in Real-Embedding, but allow the algorithm access to the same solution set as GRAPE.

One of the most promising benefits to such algorithms would be a dramatic decrease in the number of matrix exponentials that need to be calculated. As systems increase in

size, matrix exponentials become incredibly costly and contribute to increased run-times. Calculating the gradient on the tangent space minimizes the number of matrix exponentials that need to be computed by discovering the direction of steepest descent before passing through an exponential.

In summary, we have examined the topic of algorithms to calculate time-optimal control in closed quantum systems. The existing GRAPE algorithm was designed only for fidelity optimization and not temporal optimization. It can be modified to yield time-optimal results, and generally gives good solutions in exchange for a high computational cost. The work done on the Kaya-Huneault algorithm is promising for single-qubit cases, but may be for naught in the domain of multiple particles. This is the result of the heuristic evidence that the terminal surface on which the algorithm so crucially depends seems to be totally disconnected. Embedding the problem into $\mathbb{R}^\ell$ results in a violation of the Linear Independence Constraint Qualification which deprecates most contemporary constrained optimization methods.

Chapter 5

Singular Controls

We have already discussed how the Pontryagin Principle and its generalizations yield different classifications for extremals. While most of the attention thus far has been focused on computing time-optimal *bang-bang* controls, a full analysis of optimality necessitates examining the existence of singular extremals as well. We recall the results of Section 4.4 in which we found that Gradient Ascent Pulse Engineering was more successful in deriving time-optimal trajectories than the corresponding bang-bang algorithms. This may suggest that singular controls are an essential part of finding time-optimal solutions.

In affine systems with bounded control amplitudes, we have seen that all controls must be bang-bang or singular. While this is a sufficient characterization from the point of view of control theory, one can also approach the topic of singular controls from the perspective of functional analysis and sub-Riemannian geometry.

The tools and techniques for analyzing singular trajectories using the PMP is often very specific to the dynamics being considered, and the Maximum Principle does not offer many general tools to aid in such a study. Consequently, the inclusion of functional and geometric arguments is a necessary tool in order to create general statements concerning singular controls, and allows us to explore properties of systems that may not be evident by examining the problem from only a control-theoretical viewpoint.

We will begin with a review of some of the tools and techniques used in singular control theory, after which we will examine such controls in the domain of propagator evolution in closed quantum systems. This chapter relies heavily on the contents of Chapter 2 and in particular the differential geometric components introduced therein.

5.1 The Tools of Singular Control

Many of the techniques developed in sub-Riemannian geometry utilize a functional analytic approach to study mappings between a curve and its parameterization. In our case, the parameters will be controls and the corresponding curves are precisely evolution trajectories on $\mathfrak{U}(d)$. In order to begin our exploration of singular control in quantum systems, it will first be necessary to create a toolbox from which we may analyze our environment.

5.1.1 The Endpoint Mapping

The endpoint map plays a crucial role in the study of singular extremals and gives us the ability to move between a point on a curve and its corresponding parameterization given a fixed parameter domain. Depending on the context of its use there are many equivalent ways to define the endpoint mapping. We will introduce it here both in its sub-Riemannian and control theoretic framework.

Definition 5.1.1. Let $(M, \mathcal{D}, g)$ be a sub-Riemannian manifold and denote by $W_g^{1,2}([t_0, t_1])$ the space of functions γ such that both γ and $\dot{\gamma}$ are square integrable under the sub-Riemannian metric g on the domain $[t_0, t_1]$. We choose to use the notation W for the relation of this space to Sobolev spaces.

From [54], the *endpoint map* at the point $p \in M$ is the map that takes each horizontal curve beginning at p to its endpoint. If $\Omega_p([t_0, t_1], \mathcal{D})$ is the set of all horizontal curves $\gamma : [t_0, t_1] \to M$ such that $\gamma \in W_g^{1,2}([t_0, t_1])$, the endpoint mapping is the map $E_p : \Omega_p([t_0, t_1]) \to M$ acting as $E(\gamma) = \gamma(t_1)$.

In order to cast this into a control theoretical framework, let $\{H_1, \ldots, H_m\}$ be a global orthonormal frame for a set of complete vector fields H_i. If $\gamma \in W_g^{1,2}([t_0, t_1])$ then we can write

$$\dot{\gamma} = \sum_{i=1}^{m} u^i H_i \tag{5.1}$$

and the coordinates u^i give a parameterization of our curve γ.

Consider the geometric control system

$$\frac{dX}{dt} = F(X, u(X, t)) \tag{5.2}$$

for $X \in M$ and admissible control set $\mathcal{U} \in \mathbb{R}^m$. If $T > 0$, we define the endpoint mapping[11]

$$\begin{aligned} E : M \times \mathbb{R}_+ \times \mathcal{U} &\to G \\ (X_0, T, \hat{u}) &\mapsto X(T) \end{aligned}, \tag{5.3}$$

where $X(t)$ satisfies (5.2) with $u(X,t) = \hat{u}$ and $X(0) = X_0$. Let $u(t) \in L^\infty([0,t], \mathbb{R}^m)$ and give an initial condition $X_0 \in M$. Let Φ denote the flow of (5.2). The flow through X_0 parameterized by the control u is given by

$$\Phi(t, X_0, u) = \mathcal{T} \exp \left[\int_0^t F(X(t), u(X,t))\ dt \right] X_0 \tag{5.4}$$

and we can view E as a function taking a control to the endpoint of its associated flow by [5]

$$E : u \mapsto \Phi(T, X_0, u) = \mathcal{T} \exp \left[\int_0^T F(X(t), u(X,t))\ dt \right] (X_0). \tag{5.5}$$

Let $\Phi_t^{X_0}(\cdot) = \Phi(t, X_0, \cdot)$ and $\Phi_{t,u}(\cdot) = \Phi(t, \cdot, u)$. Since each control uniquely defines a flow curve $\Phi_t^{X_0}$, the sub-Riemannian endpoint mapping is equivalent to the control theoretical one when curves are parameterized by their controls.

Critical points of differentials correspond to points at which the mapping is not surjective. We will show in Section 5.1.3 that singular trajectories correspond to critical points of the endpoint mapping. For this to make sense, we need to ensure that the differential of the endpoint mapping is well defined.

Proposition 5.1.2 (Montgomery [56])**.** *Let $(M, \mathcal{D}, g)$ be a sub-Riemannian manifold and assume that $X(t) \in W_g^{1,2}([t_0, t_1])$ is generated by a set of non-autonomous vector fields $H_i(X)$ so that*

$$X(t) = \sum_{i=1}^m u^i(t) H_i(X(t)). \tag{5.6}$$

with initial condition $X(t_0) = X_0$. The differential of the endpoint mapping is given by

$$dE_X(v) = d\Phi_{t_1,u}(X_0) \int_{t_0}^{t_1} (d\Phi_{t,u}(X_0))^{-1} \left(\sum_{i=1}^m v^i(t) H_i(X(t)) \right)\ dt \tag{5.7}$$

Proof. Similar to many approaches in the calculus of variations, we first notice that $X(t) = \Phi_t^{X_0}(u)$ and introduce a control perturbation in $X(t)$ by defining

$$X_\epsilon(t) = \Phi_t^{X_0}(u + \epsilon v) = \sum_{i=1}^m \left[u^i(t) + \epsilon v^i(t) \right] H_i(X(t)). \tag{5.8}$$

Define $\delta X(t) = \frac{\partial X_\epsilon}{\partial \epsilon}\big|_{\epsilon=0}$ and notice that (5.6) gives us that

$$\frac{dX_\epsilon}{dt} = \sum_{i=1}^{m} \left[u^i(t) + \epsilon v^i(t)\right] H_i(X_\epsilon(t)). \tag{5.9}$$

Defining a fixed but arbitrary set of coordinates x, we can use the fact that $\frac{\partial}{\partial t}$ and $\frac{\partial}{\partial \epsilon}$ commute to differentiate (5.9) as

$$\frac{d}{d\epsilon}\frac{dX_\epsilon}{dt}\bigg|_{\epsilon=0} = \frac{d}{dt}\delta X(t) = \sum_{i=1}^{m} \left[v^i(t)H_i(X(t)) + u^i(t)\frac{\partial H_i}{\partial x}\delta X(t)\right]. \tag{5.10}$$

This is an inhomogeneous, linear differential equation which we can solve by the method of variation of parameters. Let $\Psi(t)$ be the fundamental matrix solution of the homogeneous equation, so that

$$\frac{d\Psi}{dt} = \left[\sum_{i=1}^{m} u^i(t)\frac{\partial H_i}{\partial x}\right]\Psi, \qquad \Psi(t_0) = \mathrm{id} \tag{5.11}$$

and assume that the solution to (5.10) is of the form $\delta X(t) = \Psi(t)w(t)$ where $w(t)$ satisfies

$$\Psi(t)\frac{d}{dt}w(t) = \sum_{i=1}^{m} v^i(t)H_i(X(t)). \tag{5.12}$$

This implies that

$$\delta X(t) = \Psi(t)\left(\int_{t_0}^{t} \Psi(s)^{-1}\left(\sum_{i=1}^{m} v^i(t)H_i(X(t))\right)\, ds\right). \tag{5.13}$$

Since $\Phi_t^{X_0}(u) = X(t)$ we have $E_X = \Phi_{t_1}^{X_0}(u)$ which gives

$$dE_X(v) = \left.\frac{\partial \Phi_{t_1}^{X_0}(u+\epsilon v)}{\partial \epsilon}\right|_{\epsilon=0} = \left.\frac{\partial X_\epsilon}{\partial \epsilon}\right|_{\epsilon=0} = \delta X(t). \tag{5.14}$$

Hence all that remains to be shown is that $\Psi(t) = d\Phi_t^{X_0}(v)$. This result follows almost immediately. Since $X(t) = \Phi_t^{X_0}(u)$ equation (5.6) implies that

$$\frac{d}{dt}\Phi_t^{X_0}(u) = \sum_{i=1}^{m} u^i(t)H_i(t, \Phi_t^{X_0}(u)) \tag{5.15}$$

and so

$$\frac{d}{dt}d\Phi_t^{X_0}(u) = \left[\sum_{i=1}^{m} u^i(t)\frac{\partial H_i}{\partial x}\right] d\Phi_t^{X_0}(u). \tag{5.16}$$

By comparing (5.11) and (5.16) we see that they satisfy the same differential equation so that $\Psi(t) = d\Phi_t^{X_0}(u)$ and the result follows. □

5.1.2 Sub-Riemannian Geometry

Our goal will be to show why critical points of the endpoint mapping correspond to singular trajectories. This will be the result of casting the sub-Riemannian structure into one that is similar to classical Hamiltonian mechanics. Before we can relate the endpoint mapping to control theory, it will be necessary to discuss some fundamental results in sub-Riemannian geometry.

It is well know that any smooth manifold G can be endowed with a Riemannian structure which simply corresponds to a non-degenerate, positive-definite section of the vector bundle $\Sigma^2 T^*G$ of symmetric 2-tensor fields on the cotangent bundle. In general, no such structure can be defined on a sub-Riemannian geometry, though we can define a similar concept.

Definition 5.1.3. A *cometric* is a section of the vector bundle $\Sigma^2 TG$ of symmetric 2-tensor fields on the tangent bundle.

In order to define the analogous property to a metric on a sub-Riemannian manifold, it is necessary to consider the dual space to that of the metric. To see how this relates to sub-Riemannian geometry, note that just as a Riemannian metric defines an inner-product on tangent space T_XG, the cometric defines a fibre-bilinear form on cotangent spaces T_X^*G. Since there is no positive-definite condition, it would be incorrect to say "inner-product." If $\beta \in \Sigma^2 T^G$ is a cometric, we define that bilinear form as

$$(\phi, \psi)_X = \beta_X(\phi, \psi), \qquad \forall \phi, \psi \in T_X^*G. \tag{5.17}$$

Such a cometric can be used to define a dual-symmetric bundle map $\beta : T^*G \to TG$ such that for all $\phi, \psi \in T_X^*G$, $(\phi, \psi)_X = \phi(\beta_X(\psi))$. This use of β to define both the cometric and the bundle map is an obvious abuse of notation, though the context of its use should make it clear as to which form we are referring. This is similar to the musical isomorphisms of Riemannian geometry and give us the same bilinear versus dual-space inner-product analogy as that discussed in Section 2.4.1.

Lemma 5.1.4 (Montgomery [56])**.** *Let G be a manifold.*

If $(G, \mathcal{D}, g)$ is a sub-Riemannian manifold, we can define a unique bundle map $\beta : T^*G \to TG$ such that

1. *The bundle map defines the distribution; that is, $\forall X \in G, \text{Image}(\beta_X) = \mathcal{D}_X$.*
2. *The bundle map agrees with the sub-Riemannian metric; that is, $\forall X \in G, \forall H \in \mathcal{D}_X$ and $\forall \phi \in T^*_X G$ we have that $\phi(H) = g_X(\beta_X(\phi), H)$.*

Conversely, any constant-rank cometric $\beta \in \Sigma^2 TG$ defines a sub-Riemannian geometry of the same rank as β by the induced bundle map.

This proposition allows us to say something that has a profound meaning both mathematically and physically.

Definition 5.1.5. Let β be a cometric on G with the bilinear form $(\cdot, \cdot)_X$ on $T^*_X G$. The fibre-quadratic function defined by $H(X, \phi) = \frac{1}{2}(\phi, \phi)_X$ is called the *sub-Riemannian Hamiltonian of G corresponding to the cometric β.*

Proposition 5.1.6 (Montgomery [56])**.** *Every sub-Riemannian structure is uniquely determined by its sub-Riemannian Hamiltonian. Conversely, any non-negative fibre-bilinear Hamiltonian of constant fibre rank k gives rise to a sub-Riemannian structure whose underlying distribution is rank k.*

This result follows from Lemma 5.1.4. A Hamiltonian is defined by its cometric which defines a sub-Riemannian structure and conversely, a sub-Riemannian structure *uniquely* defines a cometric which gives rise to a Hamiltonian.

Similar to our discussion of symplectic geometry in Section 2.3, we can begin to see the development of a state-momentum concept in which the sub-Riemannian Hamiltonian plays the role of kinetic energy. In fact, this can be extended further via the following definitions:

Definition 5.1.7. Let G be a smooth manifold and $H \in \Gamma(TG)$. The fibre-linear function $P_H : T^*G \to \mathbb{R}$ given by $P_H(X, \phi) = \phi(H(X))$ is the momentum function for H.

Proposition 5.1.8. *Let $(G, \mathcal{D}, g)$ be a sub-Riemannian geometry and $\{H_1, \ldots, H_k\}$ a frame for $\mathcal{D}$. If we define the sub-Riemannian metric in terms of this frame as*

$$g_{ij}(X) = \langle H_i(X), H_j(X) \rangle_X \tag{5.18}$$

then the sub-Riemannian Hamiltonian is described by

$$H(X,\phi) = \frac{1}{2}\sum_{i,j} g^{ij}(X) P_{H_i}(X,\phi) P_{H_j}(X,\phi) \tag{5.19}$$

where $\sum_b g^{ib} g_{bj} = \delta^i_j$.

The definition of g_{ij} given in Proposition 5.1.8 is nothing more than the sub-Riemannian metric in local coordinates. Furthermore, the condition that $\sum_b g^{ib} g_{bj} = \delta^i_j$ states that we have defined g^{ij} to be the inverse of g_{ij}.

Proof. From Lemma 5.1.4 we know there exists a bundle map $\beta : T^*G \to TG$ such that $\phi(H) = \langle \beta_X(\phi), H\rangle_X$. Let $(\cdot,\cdot)$ be the fibre-quadratic bilinear form induced by the bundle map. By the definition of this map, it follows that $(\phi,\psi)_X = \phi(\beta_X(\psi))$. Combining these, we see that

$$(\phi,\phi)_X = \phi(\beta_X(\phi)) = \langle \beta_X(\phi), \beta_X(\phi)\rangle_X . \tag{5.20}$$

Since $\{H_i\}_{i=1}^k$ is a frame for $\mathcal{D}$, we can write $\beta_X(\phi)$ in terms of the H_i as

$$\beta_X(\phi) = \sum_{i=1}^k c^i(X,\phi) H_i(X) \tag{5.21}$$

for some set of coefficients c^i that have yet to be determined. Plugging (5.21) into (5.20) yields

$$\begin{aligned}
\langle \beta_X(\phi), \beta_X(\phi)\rangle_X &= \left\langle \sum_{i=1}^k c^i(X,\phi) H_i(X), \sum_{i=1}^k c^i(X,\phi) H_i(X) \right\rangle \\
&= \sum_{i,j=1}^k c^i(X,\phi) c^j(X,\phi) \langle H_i(X), H_j(X)\rangle \\
&= \sum_{i,j=1}^k g_{ij}(X) c^i(X,\phi) c^j(X,\phi)
\end{aligned} \tag{5.22}$$

We our focus to discovering the form of the coefficients $c^i(X,\phi)$. Realize that

$$\begin{aligned}
\langle \beta_X(\phi), H_j\rangle_X &= \sum_{i=1}^k c^i(X,\phi) \langle H_i(X), H_j(X)\rangle \\
&= \sum_{i=1}^k c^i(X,\phi) g_{ij}(X)
\end{aligned} \tag{5.23}$$

but (5.23) simply corresponds to matrix-vector multiplication. By using the inverse matrix g^{ij} we can thus write

$$c^i(X,\phi) = \sum_{j=1}^{k} g^{ij}(X) \left\langle \beta_X(\phi), H_j \right\rangle_X . \tag{5.24}$$

Furthermore, by Lemma 5.1.4

$$\left\langle \beta_X(\phi), H_j \right\rangle_X = \phi(H_j(X)) = P_{H_j}(X,\phi) \tag{5.25}$$

and so we get

$$c^i(X,\phi) = \sum_{j=1}^{k} g^{ij}(X) P_{H_j}(X,\phi). \tag{5.26}$$

Finally, we can substitute (5.26) into (5.22) to get

$$\begin{aligned}
(\phi,\phi)_X &= \sum_{i,j=1}^{k} g_{ij}(X) c^i(X,\phi) c^j(X,\phi) \\
&= \sum_{i,j=1}^{k} g_{ij}(X) \left(\sum_{m=1}^{k} g^{im}(X) P_{H_m}(X,\phi) \right) \left(\sum_{n=1}^{k} g^{in}(X) P_{H_n}(X,\phi) \right) \\
&= \sum_{i,j,m,n} g_{ij}(X) g^{im}(X) g^{jn}(X) P_{H_m}(X,\phi) P_{H_n}(X,\phi) \qquad (5.27) \\
&= \sum_{m,n} g^{mn}(X) P_{H_m}(X,\phi) P_{H_n}(X,\phi) \qquad (5.28)
\end{aligned}$$

where in (5.27) we have used the fact that

$$\sum_j g^{jn} \left(\sum_i g_{ij} g^{im} \right) = \sum_j g^{jn} \delta_j^m = g^{mn}. \tag{5.29}$$

The result then follows from the definition of $H(X,\phi)$. □

5.1.3 Critical Points are Singular

We are very close to showing why the endpoint mapping is such a useful tool in singular analysis. In order to proceed further, we will need to unite the previous topics into a unified framework.

Definition 5.1.9. Let $(G, \mathcal{D}, g)$ be a sub-Riemannian manifold. We define the *annihilator of* $\mathcal{D}$ as

$$\mathcal{D}^{\perp} = \left\{ \xi \in T^*M \;\middle|\; \xi(v) = 0, \forall v \in \mathcal{D} \right\}. \tag{5.30}$$

$\mathcal{D}^{\perp}$ consists of all the covector fields for which the horizontal distribution is a subset of its kernel.

Let ω be the canonical symplectic form on T^*M and $\hat{\omega} = \omega\Big|_{\mathcal{D}^{\perp}}$ be the restriction of this form to $\mathcal{D}^{\perp}$ and note that this restriction need not be symplectic [22].

Definition 5.1.10 (Chitour [22] and Montgomery [56])**.** Let $\mathcal{D}^{\perp}$ be as defined above. A *characteristic* for $\mathcal{D}^{\perp}$ is an absolutely continuous curve $\psi : [t_0, t_1] \to \mathcal{D}^{\perp}$ that never intersects the zero-section of $\mathcal{D}^{\perp}$ and satisfies $\iota_{\dot{\psi}(t)}\hat{\omega} = 0$ on every t for which $\dot{\psi}(t)$ is defined. Alternatively, $\psi(t)$ is characteristic for $\mathcal{D}^{\perp}$ if $\dot{\psi}(t) \in \ker \hat{\omega}(\psi(t))$.

Such curves are called characteristic in the study of sub-Riemannian geometry. In the regime of control theory these are more commonly known as abnormal curves. The original work of Pontryagin [61] and later the generalization by Hsu [36] gives the following theorem:

Theorem 5.1.11. *Let $(G, \mathcal{D}, g)$ be a sub-Riemannian geometry and denote by Ω_{X_0} the set of all horizontal curves on $\mathcal{D}$ that start at $X_0 \in G$. If $\mathcal{D}^{\perp}$ is the annihilator of $\mathcal{D}$, it then follows that $\gamma \in \Omega_{X_0}$ is a singular trajectory if and only if it is the projection of a characteristic on $\mathcal{D}^{\perp}$. More precisely, $\exists \psi : [t_0, t_1] \to \mathcal{D}^{\perp}$ such that $\gamma(t) = \pi(\psi(t))$.*

This theorem gives us some computational leverage with discovering singular trajectories. It states that there is a bijective correspondence between characteristics of $\mathcal{D}^{\perp}$ and singular trajectories so it is sufficient to calculate characteristics alone. We shall take a look at how covectors in $\mathcal{D}^{\perp}$ interact with the endpoint mapping to discover why this is true.

The following argument is proposed in Montgomery's book [56], although we have filled in the details. Let $X : [t_0, t_1] \to M$ be a horizontal curve generated by a set of vector fields $u^i(t)H_i(X(t))$ with $X(t_0) = X_0$. Define the spatial flow of these vector fields for fixed $t \in [t_0, t_1]$ and control parameterization u^i as $\Phi_{t,u} : G \to G$. Now $d\Phi_{t,u} : TG \to TG$ and so the pullback acts as $(d\Phi_{t,u})^* : T^*G \to T^*G$. Let $\lambda_{t_1} \in T^*_{X(t_1)}G$ be an arbitrary covector

and define $\lambda_0 = (d\Phi_{t_1,u})^*_{X_0}(\lambda_{t_1})$. Using (5.7) we get

$$\begin{aligned}
\langle \lambda_{t_1}, dE_X(v) \rangle &= \left\langle \lambda_{t_1}, (d\Phi_{t_1,u})_{X_0} \int_{t_0}^{t_1} (d\Phi_{t,u}^{-1})_{X_0} \left(\sum_{i=1}^{m} v^i(t) H_i(X(t)) \right) \, dt \right\rangle \\
&= \left\langle (d\Phi_{t_1,u})^*_{X_0}(\lambda_{t_1}), \int_{t_0}^{t_1} (d\Phi_{t,u}^{-1})_{X_0} \left(\sum_{i=1}^{m} v^i(t) H_i(X(t)) \right) \, dt \right\rangle \\
&= \left\langle \lambda_{t_0}, \int_{t_0}^{t_1} (d\Phi_{t,u}^{-1})_{X_0} \left(\sum_{i=1}^{m} v^i(t) H_i(X(t)) \right) \, dt \right\rangle \\
&= \int_{t_0}^{t_1} \left\langle \lambda(t), \sum_{i=1}^{m} v^i(t) H_i(X(t)) \right\rangle \, dt
\end{aligned} \tag{5.31}$$

where we have defined $\lambda(t) = (d\Phi_{t,u}^{-1})^*_{X_0}(\lambda_0)$.

This tells us that the dual action of covectors on the endpoint differential gives a covector $\lambda(t)$ that acts opposite the generating vector fields. Any such curve $(X(t), \lambda(t))$ is the solution of a Hamiltonian system with initial condition (X_0, λ_0) and Hamiltonian

$$H_u(q, p, t) = \sum u_i(t) P_{X_i}(q, p) \tag{5.32}$$

where $P_{X_i}(q, p)$ is the usual momentum function given by $P_{X_i}(q, p) = \langle p, X_i(q) \rangle = p(X_i(q))$ is the momentum function.

We notice that (5.32) is precisely the Pontryagin Hamiltonian. Furthermore, let us evaluate the integrand of (5.31) more thoroughly

$$\left\langle \lambda(t), \sum_{i=1}^{m} v^i(t) H_i(X(t)) \right\rangle = \sum_{i=1}^{m} v^i(t) \left\langle \lambda(t), H_i(X(t)) \right\rangle = \sum_{i=1}^{m} v^i(t) P_{X_i}(X(t), \lambda(t)). \tag{5.33}$$

Putting this back into (5.31) we get

$$\langle \lambda, dE_X(v) \rangle = \int_{t_0}^{t_1} \sum_{i=1}^{m} v^i(t) P_{X_i}(X(t), \lambda(t)) \tag{5.34}$$

so that the adjoint of the endpoint differential gives us

$$(dE_X)^*(\lambda) = (P_1(t), \ldots, P_m(t)) \tag{5.35}$$

where $P_i(t) = P_{X_i}(X(t), \lambda(t))$.

Assume that γ is a critical point of the endpoint map $E : \Omega_{X_0}([t_0, t_1]) \to G$, so that the pushforward of E given by $dE : T\Omega_{X_0}([t_0, t_1]) \to TG$ is not surjective. For a fixed curve X, this means that the image of dE_X is a strict subset of the endpoint tangent space. Mathematically, we have

$$\text{Image}(d(E_X)) \subsetneq T_{X(t_1)}M. \tag{5.36}$$

The lack of surjectivity implies that we can find a non-trivial covector in the annihilator frame; that is, $\exists \lambda \in T^*_{X(t_1)}M$ such that $\forall w \in \text{Image}(dE_X)$, $\lambda(w) = 0$. Since $\omega \in \text{Image}(dE_X)$ we can find an element in its preimage, say $v \in M$, that satisfies $dE_X(v) = w$ and so $\lambda(dE_X(v)) = 0$. We chose w arbitrarily which corresponds to an arbitrary choice of preimage element v, so (5.35) implies that the only way $\lambda(dE_X(v)) = 0$ for all v is if $P_{X_i}(X(t), \lambda(t)) \equiv 0$ for each $i = 1, \ldots, m$. In a control theoretical framework, the switching functions are precisely the $P_{X_i}(X(t), \lambda(t))$ and so we see that the two definitions of singular curves are actually equivalent.

5.2 Singular Controls In Quantum Mechanics

The few papers that consider the existence of singular controls are primarily dedicated to very specific systems and often include dissipative effects [13, 15, 51, 71]. Work done by [73] has considered singular quantum control, though has not elaborated upon any results beyond those found in the text [12]. We present here a brief treatment of a general but restricted class of singular controls in quantum mechanics, from which the techniques are taken principally from [12].

5.2.1 Poisson Brackets and Manifolds

The Poisson bracket commonly appears in the study of classical mechanics and has a deep relation to the Lie bracket. It allows us to define another natural Lie-algebra on any smooth manifold, and is related to the usual Lie algebra whenever that manifold is also a Lie group.

Definition 5.2.1. Let G be a smooth manifold, $f, g \in C^\infty(T^*G)$. Denote by $\vec{f}$ the Hamiltonian vector field of f and let ω be the natural symplectic form. We define the *Poisson bracket* $\{f, g\}$ in any one of the following equivalent ways

$$\{f, g\} = \vec{f}g = \langle df, \vec{g} \rangle = \omega\left(\vec{f}, \vec{g}\right) = \left.\frac{d}{dt}\right|_{t=0} f \circ (\exp t\vec{g}). \tag{5.37}$$

Proposition 5.2.2. *Let G be a smooth manifold and fix a Darboux coordinate system (x^i, p_i) in a neighbourhood of $p \in G$. Then for $f, g \in C^\infty(T^*G)$ we have*

$$\{f, g\} = \sum_i \frac{\partial f}{\partial x^i}\frac{\partial g}{\partial p_i} - \frac{\partial f}{\partial p_i}\frac{\partial g}{\partial x^i} \tag{5.38}$$

Proof. Let $f, g \in C^\infty(T^*G)$ and recall from (2.47) that the Hamiltonian vector field of f can be written in a neighbourhood of p in Darboux coordinates as

$$\vec{f} = \sum_{i=1}^{n} \left(\frac{\partial f}{\partial x^i}\frac{\partial}{\partial p_i} - \frac{\partial f}{\partial p_i}\frac{\partial}{\partial x^i} \right) \tag{5.39}$$

Since the Poisson bracket can be defined as $\{f, g\} = \vec{f}g$, the result follows immediately. □

Proposition 5.2.3. *Let G be a Lie group and $X, Y \in \Gamma(TM)$ be smooth vector fields. If P_X, P_Y represent the corresponding momentum functions $P_X(q, p) = p(X_q)$ then*

$$\{P_X, P_Y\} = -P_{[X,Y]}. \tag{5.40}$$

Proof. We first need to examine what the momentum function looks like in local coordinates. Let (x^i) be a local frame and write

$$X_q = \sum_i X^i(q) \left.\frac{\partial}{\partial x^i}\right|_q, \qquad Y_q = \sum_i Y^i(q) \left.\frac{\partial}{\partial x^i}\right|_q. \tag{5.41}$$

For $(q, p) \in T^*G$ we have

$$P_X(q, p) = p(X_q) = \sum_i X^i(q) p_i(q), \qquad p_i(q) = P_{\frac{\partial}{\partial x^i}\big|_q}. \tag{5.42}$$

The p_i are simply coordinate momentum functions and in fact (x^i, p_i) are a Darboux coordinate systems. Using 5.38 we get

$$\begin{aligned} \{P_X, P_Y\}(q, p) &= \sum_{i,j} \left\{ X^i(q) p_i(q), Y^i(q) p_j(q) \right\} \\ &= \sum_{i,j} Y^j(q) \frac{\partial X^i}{\partial q^j} p_i(q) - X^i(q) \frac{\partial Y^j}{\partial q^i} p_j(q). \end{aligned} \tag{5.43}$$

Conversely, we recall from (2.18) that

$$[X,Y]_q = \sum_i \left(X^i(q)\frac{\partial Y^j}{\partial q^i} - Y^i(q)\frac{\partial X^j}{\partial q^i} \right) \frac{\partial}{\partial q^i} \tag{5.44}$$

so that

$$-P_{[X,Y]_q} = \sum_{i,j} \left(Y^i(q)\frac{\partial X^j}{\partial q^i} - X^i(q)\frac{\partial Y^j}{\partial q^i} \right) p_i(q) \tag{5.45}$$

Changing the dummy indices in (5.43) and comparing to (5.45) shows that these are indeed identical, as required. □

Proposition 5.2.3 gives us a powerful tool for relating Lie bracket and Poisson brackets, and is integral to the proof of the following theorem.

Theorem 5.2.4 (Properties of the Poisson Bracket, Agrachev [5])**.** *Let $f, g \in C^\infty(T^*M)$. Then*

1. $\{f,g\} = -\{g,f\}$
2. $\overrightarrow{\{f,g\}} = [\vec{f},\vec{g}]$
3. $\{\{f,g\},h\} + \{\{g,h\},f\} + \{\{h,f\},g\} = 0$

Proof. Let $f, g, h \in C^\infty(T^*M)$ and ω be the canonical symplectic form.

1. Using the symplectic definition of the Poisson bracket, we have that
$$\{f,g\} = \omega\left(\vec{f},\vec{g}\right) = -\omega\left(\vec{g},\vec{f}\right) = -\{g,f\}.$$

2. This is an exercise in applying the definition of Hamiltonian vector fields as well as the properties of Lie derivatives to which we refer the reader to Theorem 2.2.23.
$$\begin{aligned}\overrightarrow{\{f,g\}} \lrcorner \omega &= -d\{f,g\} = -d\left(\vec{f}g\right) = -d\left(\mathcal{L}_{\vec{f}}g\right) = -\mathcal{L}_{\vec{f}}dg \\ &= -\mathcal{L}_{\vec{f}}(\vec{g}\lrcorner\omega) = -\left(\mathcal{L}_{\vec{f}}\vec{g}\right)\lrcorner\omega - \underbrace{\vec{g}\lrcorner\mathcal{L}_{\vec{f}}\omega}_{=0} \\ &= -[\vec{g},\vec{f}]\lrcorner\omega = [\vec{f},\vec{g}]\lrcorner\omega.\end{aligned}$$
Since ω is non-degenerate, this implies that these two quantities are defined by their action on the contraction and hence are equal, so $\overrightarrow{\{f,g\}} = -[\vec{f},\vec{g}]$ as required.

3. Here we make explicit use of the result proven in 2. Indeed

$$\begin{aligned}\{\{f,g\},h\} &= \overrightarrow{\{f,g\}}h = [\vec{f},\vec{g}]h = \vec{f}\vec{g}h - \vec{g}\vec{f}h \\ &= \vec{f}\{g,h\} - \vec{g}\{f,h\} = \{f,\{g,h\}\} - \{g,\{f,h\}\} \\ &= -\{\{g,h\},f\} - \{\{h,f\},g\}\end{aligned}$$

where in the last line we used the anti-symmetric property of the Poisson bracket.

□

This theorem tells us that we can endow the cotangent bundle with a Poisson bracket to get a Lie algebra. Mathematically, if G is a smooth manifold then $(C^\infty(T^*G), \{\cdot,\cdot\})$ is a Lie-algebra. The mapping given by Proposition 5.2.3 is a Lie-algebra anti-homomorphism between Lie(G) and $(C^\infty(T^*G), \{\cdot,\cdot\})$.

There may be instances in which it is more natural to work with the Poisson bracket instead of the Lie bracket, but the above corollary indicates that the two are related. In the following analysis, we will provide characterizations in terms of both brackets.

5.2.2 Single-Input Affine Systems

Let us consider a single-input, affine control system with drift. The simplicity of this system allows us to consider the singular controls in an abstract way without needing to explicitly specify the space in which we are working. We shall first work in a very abstract sense and then specialize to the case of quantum mechanics.

Let G be a smooth Lie group with Lie algebra $\mathfrak{g}$ and consider a control system of the form

$$\dot{X}(t) = H_0(X(t)) + u(t)H_1(X(t)), \tag{5.46}$$

where $X : [t_0, t_1] \to M$ and $H_0, H_1 \in \Gamma(TM)$. For simplicity of notation and calculation, we shall denote by $\langle p, H_i\rangle$ the evaluation $p(H_i)$. Consider a time-optimal cost function, so that applying the maximum principle, we find that the Pontryagin Hamiltonian is given by

$$H_{p_0}(X,p,u) = p_0 + \langle p(t), H_0(X(t))\rangle + u(t)\langle p(t), H_1(X(t))\rangle \tag{5.47}$$

for which a singular extremal must satisfy $\langle p, H_1(X)\rangle \equiv 0$.

Proposition 5.2.5. *The adjoint system for* (5.46) *evolves as*

$$\dot{p}(t) = -p(t)\left(\frac{\partial H_0}{\partial X}(X(t)) + u(t)\frac{\partial H_1}{\partial X}(X(t))\right) \tag{5.48}$$

$$= -\left\langle p(t), \frac{\partial H_0}{\partial X}(X(t)) + u(t)\frac{\partial H_1}{\partial X}(X(t))\right\rangle \tag{5.49}$$

Proof. The proof is trivial and follows from Hamilton's equation

$$\dot{p}(t) = -\frac{\partial H_{p_0}}{\partial X}.$$

□

If we differentiate the mapping $t \mapsto \langle p(t), H_1(X(t))\rangle$ and use (5.48) we find that

$$\begin{aligned}
\frac{d}{dt}\langle p(t), H_1(X(t))\rangle &= \langle \dot{p}, H_1(X)\rangle + \left\langle p, \frac{\partial H_1}{\partial X}\dot{X}\right\rangle \\
&= \left\langle -p\left(\frac{\partial H_0}{\partial X} + u\frac{\partial H_1}{\partial X}\right), H_1(X)\right\rangle + \left\langle p, \frac{\partial H_1}{\partial X}\left(H_0(X) + uH_1(X)\right)\right\rangle \\
&= \left\langle p, -H_1(X)\left(\frac{\partial H_0}{\partial X} + u\frac{\partial H_1}{\partial X}\right)\right\rangle + \left\langle p, \frac{\partial H_1}{\partial X}\left(H_0(X) + uH_1(X)\right)\right\rangle \\
&= \left\langle p, \left(\frac{\partial H_1}{\partial X}H_0(X) - H_1(X)\frac{\partial H_0}{\partial X}\right) + u\left(\frac{\partial H_1}{\partial X}H_1(X) - H_1(X)\frac{\partial H_1}{\partial X}\right)\right\rangle \\
&= \langle p, [H_1, H_0](X) + u\underbrace{[H_1, H_1]}_{=0}(X)\rangle \\
&= \langle p(t), [H_1, H_0](X(t))\rangle
\end{aligned} \tag{5.50}$$

Since $\langle p(t), H_1(X(t))\rangle = 0$ then certainly $\frac{d}{dt}\langle p(t), H_1(X(t))\rangle = 0$. Substituting this into (5.50) it follows that $\langle p(t), [H_1, H_0](X(t))\rangle = 0$ as well. Proceeding in the same manner, we differentiate the mapping $t \mapsto \langle p(t), [H_1, H_0](X(t))\rangle = 0$ again to get

$$\langle p(t), [[H_1, H_0], H_0](X(t))\rangle + u(t)\langle p(t), [[H_1, H_0], H_1](X(t))\rangle = 0 \tag{5.51}$$

which allows us to solve for $u(t)$ explicitly as

$$u(t) = -\frac{\langle p(t), [[H_1, H_0], H_0](X(t))\rangle}{\langle p(t), [[H_1, H_0], H_1](X(t))\rangle}. \tag{5.52}$$

We introduce a common notation that allows us to write this in a very succinct manner. Define $\operatorname{ad}_X^0 Y = X$ and recursively define $\operatorname{ad}_X^k Y = [\operatorname{ad}_X^{k-1}, Y]$ for $k \geq 2$. Using the anti-symmetry of the Lie bracket, we can write

$$[[H_1, H_0], H_0] = [H_0, [H_0, H_1]] = \operatorname{ad}_{H_0}^2 H_1$$
$$[[H_1, H_0], H_1] = -[H_1, [H_1, H_0]] = -\operatorname{ad}_{H_1}^2 H_0$$

so that the control scheme given by (5.52) is

$$u(t) = \frac{\left\langle p(t), \operatorname{ad}_{H_0}^2 H_1(X(t)) \right\rangle}{\left\langle p(t), \operatorname{ad}_{H_1}^2 H_0(X(t)) \right\rangle}. \tag{5.53}$$

It is now a very simple matter to cast this into the Hamiltonian formalism which makes use of the Poisson brackets. Let define the "switching functions" as

$$H_{H_0}(X, p) = \langle p, H_0(X) \rangle, \qquad H_{H_1}(X, p) = \langle p, H_1(X) \rangle. \tag{5.54}$$

Since H_0, H_1 are vector fields, these switching functions correspond to the Hamiltonians on T^*G or equivalently, the conjugate momentum functions of H_0 and H_1. By applying Proposition 5.2.3 we see that $\langle p, [H_0, H_1](X) \rangle = \{H_{H_0}, H_{H_1}\}(X, p)$ and (5.52) becomes

$$u(t) = -\frac{\{\{H_{H_1}, H_{H_0}\}, H_{H_0}\}(X, p)}{\{\{H_{H_1}, H_{H_0}\}, H_{H_1}\}(X, p)}. \tag{5.55}$$

What we notice here is that a finite number of differentiations eventually resulted in the control term appearing linearly, allowing us to solve for it explicitly. This motivates the following definition:

Definition 5.2.6. Consider the control system previously described. For any singular trajectory $(X, p, u) : [t_0, t_1] \to T^*M \times \mathbb{R}$ define the set

$$\mathcal{R}(X, p, u) = \left\{ t \in [t_0, t_1] \;\middle|\; \{\{H_{H_0}, H_{H_1}\}, H_{H_1}\} = 0 \right\} \tag{5.56}$$

This set is possibly empty and is always an open subset of $[t_0, t_1]$. Any singular extremal for which $\mathcal{R}(X, p, u)$ is a dense in $[t_0, t_1]$ is said to be of *minimal order*.

Intuitively speaking, minimal order singular trajectories only require a finite number of differentiations on their switching functions before the control appears explicitly and linearly[14]. This allows us to solve for the control function explicitly and define a feedback controller.

5.2.3 Propagator Evolution

We want to be able to determine when a control function yields a singular trajectory, and if so whether that trajectory is optimal or not. We begin by specializing the generic results found in Section 5.2.1 to the case of quantum mechanics. In particular, our single-input affine system is of the form

$$\frac{d}{dt}X(t) = -i\left(H_0 + u(t)H_1\right)X(t), \qquad X(t_0) = \mathrm{id}. \tag{5.57}$$

In a temporary abuse of notation, if we compare this to (5.46) we see that

$$H_0(X(t)) = -iH_0X(t), \qquad H_1(X(t)) = -iH_1X(t) \tag{5.58}$$

and the control form of $u(t)$ given by (5.53) is very similar, giving

$$u(t) = \frac{\left\langle p(t), \mathrm{ad}_{H_0}^2 H_1 \right\rangle}{\left\langle p(t), \mathrm{ad}_{H_1}^2 H_0 \right\rangle} X(t). \tag{5.59}$$

Our issue is that we do not have an analytic form for $p(t)$ and hence cannot explicitly describe $u(t)$. On the other hand, knowing that $p(t)$ satisfies Hamilton's equation, we can use numerics to establish a two-point boundary value problem and iteratively solve for a singular control

The endpoint mapping gives us another tool for characterizing when controls will be singular. Recall that the differential of the endpoint mapping was given by

$$dE_X(v) = d\Phi_{t_1}^{X_0}(v) \int_{t_0}^{t_1} d\Phi_t^{X_0}(v)^{-1} \left(\sum_{i=1}^{m} v^i(t) H_i(X(t)) \right) \, dt. \tag{5.60}$$

By comparing (5.16) to (5.57) we see that

$$d\Phi_t^{X_0}(v) = \mathcal{T} \exp\left[-i(H_0 + vH_1)t\right] X_0 = X(t) \tag{5.61}$$

and so by simply substituting, (5.60) becomes

$$dE_x(v) = X(t_1) \int_{t_0}^{t_1} v(t) X^\dagger(t_1) H_1 X(t) \, dt. \tag{5.62}$$

Given a fixed control function $v(t)$, the quantity $X(t_1)$ is constant, and so to determine if $v(t)$ is singular we need only check the that the integral is rank-deficient. Furthermore,

since H_1 and $X(t)$ are square and have matching dimensions, $v(t)$ is a singular control if and only if

$$\det \int_{t_0}^{t_1} v(t) X^\dagger(t_1) H_1 X(t)\ dt = 0. \tag{5.63}$$

Unfortunately, (5.63) is not very amenable to analysis and future work needs to be done to see if we can infer any information from this equation.

Chapter 6

Concluding Remarks

6.1 Summary

The objective of this thesis was to examine the problem of finding time-optimal controls to drive an initial unitary operator X_0 to an objective operator X_d with dynamics governed by the Schrödinger equation. In doing so, it was necessary to introduce a great deal of foundational material in quantum mechanics, control theory, differential geometry, and numerical analysis. We then analyzed how the theoretical work in each of these fields could be consolidated in order to explore the objective of time-optimal control. We focused our attention on the existence of singular controls and numerical algorithms for finding bang-bang controls.

Chapter 2 recapitulated the important results in control theory, especially as it pertains to time-optimal synthesis. There we analyzed how the Pontryagin principle induced a classification on extremals under bounded controls, and introduced the differential geometry necessary to extend the principle to general manifolds. Symplectic manifolds and sub-Riemannian geometries were also discussed, as they play an important role in understanding not only how a control problem is generalized, but give insight as to some of the more esoteric constructs given in the original maximum principle. These tools were especially useful in the following chapters to construct numerical algorithms and analyze the existence of singular controls.

Chapter 3 provided a brief description of the existing literature in quantum control, including issues of controllability, Cartan decompositions, and landscape topology. Therein it was found that much of the work already established in geometric control theory could be

translated into a quantum framework, and that the additional structure afforded by considering quantum systems meant that researchers could strengthen many of those generic results.

Our discussion in Chapter 4 focused on numerically computing time-optimal solutions under the assumption that no singular control existed, permitting us to consider only bang-bang solutions. We analyzed the theoretical framework and motivation established by Huneault in his thesis [37], and created our own algorithm based off of this algorithm. These algorithms were then computationally implemented and compared to a modified version of the Gradient Ascent Pulse Engineering (GRAPE) algorithm designed to find time-optimal solutions. GRAPE was able to find excellent solutions, but with extreme computational costs. Our algorithm was competitive in low dimensions, providing good solutions relatively quickly, but had trouble as the dimension of the problem started to grow. Ultimately, GRAPE consistently yielded the best time, though with high computational times.

Finally, Chapter 5 discusses singular controls. Here we introduced the endpoint mapping often used in sub-Riemannian geometry and examined its properties as a functional. We derived singular feedback control laws for single-input affine systems in terms of both the Poisson and Lie bracket formalisms. After showing the critical points of the endpoint mapping correspond to singular controls, we then used its differential to give a criterion under which controls could be singular.

6.2 Future Work

We mentioned previously that the results of Chapter 4 indicate that while the bang-bang algorithms are able to find competitive solutions to those given by GRAPE, an increase in dimension can start to cause troubles.

We first consider the fact that the violation of the Linear Independence Constraint Qualification caused the problem statement of the Real-Embedding algorithm to be degenerate. Even by demanding that the constraint set only hold up to an ϵ-tolerance, the degeneracy could not be completely removed. Hence the real embedding algorithm might perform better if it were formulated in a framework in which it is non-degenerate.

The terminal surface shared by both the Kaya-Huneault and Real-Embedding algorithms needs to be examined more closely on a theoretical level. In particular, one would like to show that the pre-image of the set of time-intervals which correspond to exact objective evolution correspond to a completely disconnected set. It would be sufficient to show that the Hessian of the function $d(X(\xi), X_d)^2$ is non-degenerate whenever ξ drives $X(\xi)$ to

precisely X_d, since Morse's Lemma would then imply all critical points are isolated. This would imply that continuous optimization methods on the terminal surface are intractable and the approaches of the Kaya-Huneault and Real-Embedding algorithms cannot be used.

Applying standard optimization algorithms in Euclidean space causes a great deal of trouble in quantum control computations, with examples including the exponential growth of classical representations of quantum systems, as well as the degeneracy of the constrained optimization problem mentioned above. In Section 4.3.1 we used a composition between a manifold curve and a real-valued map to abstract away the intermediate details of the system evolution. By instead considering how optimization processes can be performed on this intermediate geometric space, we may be able to find better algorithms. Such geometric algorithms can analytically exploit the natural structure of the space on which the evolution is occuring, resulting in more efficient computations that avoid the computational overhead of transfering between Euclidean spaces. Some theoretical foundations for this work are provided in Appendix D.

There is much work to be done in the area of singular controls. Typically, in order to consider singular controls one examines the details of a specific system. In order to generalize these results we are forced to use functional analysis and differential geometry. We have given a condition on the controls for which a trajectory will be singular, but it is not computationally friendly and needs to be worked with further. Possible avenues for exploration are the use of higher-order necessary conditions to find restrictions on the form of singular controls that might lead to more general results.

Appendix A

Differential Geometry

A.1 Tangent Spaces

Herein we will note some of the important properties of the abstract definition of a tangent space on a manifold and how it is equivalent to the case when our manifold is $\mathbb{R}$. Recall that we had defined tangent vectors as derivations as follows:

Definition A.1.1. Let M be a smooth manifold. A linear map $X : C^\infty(M) \to \mathbb{R}$ is a derivation at $p \in M$ if

$$X(fg) = f(p)Xg + g(p)Xf, \qquad \forall f, g \in C^\infty(M)$$

We note the set of all derivations at p by T_pM and refer to this as the tangent manifold at p.

While this is the preferred way in much of the more modern differential geometry literature, there are many other ways. The following is the definition using germs:

Definition A.1.2. Let M be a smooth manifold. A *smooth function element* is a couple (f, U) with $f \in C^\infty(M, \mathbb{R})$ and $U \subseteq M$ such that $f : U \to real$.

Let $p \in M$ and consider the set of smooth function elements (f, U) such that $p \in U$, and denote this set by $\mathcal{G}_p$. We define an equivalence relation on $\mathcal{G}_p$ by saying that

$$(f, U) \sim (g, V) \qquad \text{if} \qquad \exists W \ni p, \forall x \in W, f(x) = g(x) \tag{A.1}$$

Definition A.1.3. Let M be a smooth manifold and take $p \in M$. If $\mathcal{G}_p$ and $\sim$ are defined as above, and $(f, U) \in \mathcal{G}_p$ then the *germ of f at p* is the equivalence class

$$[(f, U)]_\sim \in \mathcal{G}_p / \sim \tag{A.2}$$

We will denote the set of all germs at p by $C^\infty_p = \mathcal{G}_p / \sim$.

We can define an $\mathbb{R}$-algebra on C^∞_p by defining the following operations

$$[(f, U)]_\sim + [(g, V)]_\sim = [(f + g), U \cap V]_\sim \tag{A.3}$$

$$c[(f, U)]_\sim = [(cf, U)]_\sim, \qquad c \in \mathbb{R} \tag{A.4}$$

$$[(f, U)]_\sim [(g, V)]_\sim = [(fg), U \cap V]_\sim \tag{A.5}$$

Definition A.1.4. For a smooth manifold M and $p \in M$ define the *tangent space at p* as

$$T_pM = \left\{ X : C^\infty_p \to \mathbb{R} \;\middle|\; X[(f, U)]_\sim [(g, V)]_\sim = f(p)X[(g, V)]_\sim + g(p)X[(f, U)]_\sim \right\} \tag{A.6}$$

This is very similar to our definition of derivations on functions of M, since we still require that the tangent vectors satisfy the Leibniz rule.

However, since control theory is more interested in curves on manifolds and their tangents, the following is an intuitive and alternate way of defining T_pM. Let M be a smooth manifold and $p \in M$. Consider the set of all smooth curves $\gamma : J_\gamma \subseteq \mathbb{R} \to M$ such that $\gamma(0) = p$. Then we define the following equivalence relation on this set as follows:

$$\gamma_1 \sim \gamma_2 \qquad \text{if and only if} \qquad \forall f \in C^\infty(M), (f \circ \gamma_1)'(0) = (f \circ \gamma_2)'(0) \tag{A.7}$$

That is, two curves are equivalent at p if their derivatives at p agree regardless of how we map them onto $\mathbb{R}$. Then the tangent space is the set of all equivalence classes under this relation.

All of these definitions of the tangent space are equivalent, as is well known throughout the field of differential geometry.

Proposition A.1.5. *For any $p \in \mathbb{R}^n$ we have that $T_p\mathbb{R}^n \cong \mathbb{R}^n$.*

Proof. Consider the set $\mathbb{R}^n_p = \left\{ (p, x) \;\middle|\; x \in \mathbb{R}^n \right\}$. Clearly $\mathbb{R}^n_p \cong \mathbb{R}$ under the projection $\pi_2 : \{p\} \times \mathbb{R}^n \to \mathbb{R}^n$. Now for each $(p, v) \in \mathbb{R}^n_p$, define the directional derivative mapping $D_v|_p : C^\infty(\mathbb{R}^n) \to \mathbb{R}$ by

$$D_v|_p f = \left. \frac{d}{dt} \right|_{t=0} f(p + tv) \tag{A.8}$$

We notice that $D_v|_p$ corresponds to derivations using Definition A.1.1 and hence

$$T_p\mathbb{R}^n = \left\{ D_v|_p \;\middle|\; v \in \mathbb{R}^n \right\} \tag{A.9}$$

Claim: The mapping $(p, v) \to D_v|_p$ is an isomorphism from $\mathbb{R}^n_p \to T_p\mathbb{R}^n$.

This result is proven in [52], and is long and unenlightening so we omit its proof. Consequently, we have arrived at the desired result since $\mathbb{R}^n \cong \mathbb{R}^n_p \cong T_p\mathbb{R}^n$ as required. □

A.2 Differential Forms and Tensors

Definition A.2.1. Suppose that $V_1, \ldots, V_k$ and W are complex vector spaces. A map $F : V_1 \times \ldots \times V_k \to W$ is *multilinear* if it is linear in each component with the others held constant.

This definition gives rise to two commonly used terms.

Definition A.2.2. If V and W are vector spaces and $F : V \times V \to W$, we say that F is *bilinear* if it is linear in both components. We say F is *sesquilinear* if it is linear in one component and conjugate linear in the other. More precisely, a sesquilinear function acts on scalar multiplication as follows: If $x, y \in V$ and $a, b \in \mathbb{C}$ then

$$F(ax, by) = a\bar{b}F(x, y) \tag{A.10}$$

Definition A.2.3. A covariant k-tensor T on a vector space V is a real-valued multilinear function of k-elements of V, $T : V^k \to \mathbb{R}$. We say that the rank of T is k and write $\operatorname{rank} T = k$. The set of all k-tensors on V is denoted $T^k(V)$ and is a vector space under the usual operations of pointwise addition and scalar multiplication.

Note that linear functionals $T : V \to \mathbb{R}$ are rank-1 covariant tensors, so we can identify $V^* = T^1(V)$. Our next goal is to define the set of contravariant k-tensors. This is normally done via category theoretical devices involving free vector spaces and tensor product spaces. Since these will not be useful to our study, we instead take them as given, referring the interested reader to [31] for more details.

Definition A.2.4. The set of contravariant k-tensors is defined as $T_k(V) = V^{\otimes k}$. For $k, \ell \in \mathbb{N}$, the space of mixed (k, ℓ)-tensors is $T^k_\ell(V) = (V^*)^{\otimes k} \otimes V^{\otimes \ell}$.

Since we know that the tangent and cotangent spaces have a natural vector space structure, we can extend these notions to differential geometry.

Definition A.2.5. Let M be a smooth manifold and define the mixed tensor bundle

$$T^k_\ell M = \bigsqcup_{p\in M} T^k_\ell(T_pM). \tag{A.11}$$

The smooth sections of tensor bundles are tensor fields, and we denote the set of all such sections by $\Gamma(T^k_\ell M)$.

There are two very important classes of tensors that exhibit particularly favorable symmetries. First, let $\mathcal{S}_k$ denote the symmetric group on k-letters. For the sake of notation, if $\sigma \in \mathcal{S}_k$ denote by ${}^\sigma T$ the action

$${}^\sigma T(X_1, \ldots, X_k) = T(X_{\sigma(1)}, \ldots, X_{\sigma(k)}). \tag{A.12}$$

The choice to put the permutation to the left comes from the group composition, since we observe that if $\tau \in \mathcal{S}_k$ then ${}^\tau({}^\sigma T) = {}^{\tau\sigma} T$.

Definition A.2.6. A covariant k-tensor T is symmetric if its value is unchanged by permuting two elements. Since any permutation can be written as a product of transpositions, T is symmetric if and only if $\forall \sigma \in \mathcal{S}_k, {}^\sigma T = T$. We denote the set of all symmetric k-tensors by $\Sigma^k(V)$.

Definition A.2.7. A covariant k-tensor is alternating if $\forall \sigma \in \mathcal{S}_k$ we have that $T = (\operatorname{sgn}\sigma)^\sigma T$. Note that equivalently, T is alternating if it is always zero on a linearly dependent, and switches signs on transpositions. We may call T a k-covector and denote the set of all alternating k-tensors as $\Lambda^k(V)$.

Definition A.2.8. Define the alternating k-bundle of M as

$$\Lambda^k M = \bigsqcup_{p\in M} \Lambda^k(T_pM)$$

A section of $\Lambda^k M$ is called a differential k-form.

The use of k-forms in differential geometry is quite extensive, ranging from defining volume forms to extending the notion of an exterior derivative. However, as this thesis does not make extensive use of such forms, we will omit any further discussion of them here.

Appendix B

Group and Lie Group Theory

Here we will review some of the basics of group theory as they are used in this thesis.

B.1 Group Theory

Definition B.1.1. Let G be a set $\cdot : G \times G \to G$ be a binary operator. We say that $(G, \cdot)$ is a *group* if

1. $\forall x, y, z \in G,\ (x \cdot y) \cdot z = x \cdot (y \cdot z)$.
2. $\exists e \in G,\ \forall x \in G, x \cdot e = x$.
3. $\forall x \in G,\ \exists y \in G, x \cdot y = e$.

Definition B.1.2. Let $(G, \cdot)$ be a group and X a set. A *left group action* of G on X is a binary function $f : G \times X \to X$ such that

1. $f(g \cdot h, x) = f(g, f(h, x))$
2. $f(e, x) = x$ where e is the group identity.

A *right group action* is symmetrically defined.

Definition B.1.3. Let $(G, \cdot_G)$ and $(H, \cdot_H)$ be groups. A *group homomorphism* is a function $f : G \to H$ such that $\forall p, q \in G$

$$f(p \cdot_G q) = f(p) \cdot_H f(q) \tag{B.1}$$

If $f : G \to H$ is a homomorphism, then f is an *isomorphism*. An isomorphism from G to itself is an *automorphism*.

If $p \in G$ is a fixed element, then $f_p(x) = p^{-1}xp$ is an automorphism, and is called an *inner-automorphism*.

B.2 Lie Group Theory

Definition B.2.1. In the event that G and H are Lie groups, a *Lie group homomorphism* is a function $f : G \to H$ that is a smooth group homomorphism. All subsequent terms from Definition B.1.3 follow with the added restriction of smoothness.

Proposition B.2.2. *[39] The Lie algebra of left-invariant vector fields is anti-isomorphic to the Lie algebra of right-invariant vector fields.*

Proof. Let G be a Lie group and consider the Lie group automorphism $\Phi : G \to G$ given by $G(X) = X^{-1}$. Since the identity $e \in G$ is self-inverse, the pushforward of Φ acts as $\Phi_* : T_eG \to T_eG$.

Now, if L_X is the group action of left-translation by $X \in G$, we see that

$$\Phi R_{X^{-1}} \Phi^{-1}(Y) = \Phi R_{X^{-1}} Y^{-1} = \Phi Y^{-1} X^{-1} = XY = L_X(Y) \tag{B.2}$$

so we conclude that $\Phi R_{X^{-1}} \Phi = L_X$. Applying the pushforward to this equation, we get that $\Phi_*(R_{X^{-1}}) = (L_X)_* \Phi_*$. Let H_R be a right-invariant vector field and define $H_L = \Phi_* H_R \circ \Phi^{-1}$. Then

$$\begin{aligned}(L_X)_* H_L(e) &= (L_X)_* \Phi_* H_R \Phi^{-1}(e) = \Phi_*(R_{X^{-1}}) H_R(e) \\ &= \Phi_* H_R(\Phi^{-1}(X)) = H_L(X)\end{aligned}$$

This implies that H_L is a left-invariant vector field, and so the relation

$$X_R \mapsto \Phi_* H_R \Phi^{-1} \tag{B.3}$$

is the necessary anti-isomorphism. □

Proposition B.2.3. *If G is a Lie group under juxtaposition and $F : \mathbb{R} \to G$ is a one-parameter subgroup, then* $\text{Image}(G) \leq G$.

Proof. For brevity, denote $H = \text{Image}(G)$. We first check that the binary operator is well defined. Let $h_1, h_2 \in H$, so that by definition $\exists t_1, t_2 \in \mathbb{R}$ such that $F(t_i) = h_i, i = 1, 2$. But

$$F(t_1 + t_2) = F(t_1)F(t_2) = h_1 h_2 \tag{B.4}$$

and so $h_1 h_2 \in H$.

All other properties follow almost immediately from the fact that F is a Lie-group homomorphism. Indeed H has an identity element $F(0)$ since identity elements must map to identity elements. If $h_i \in H$ and $t_i \in \mathbb{R}$ such that $F(t_i) = h_i, i = 1, \ldots, 3$ then

$$\begin{aligned}(h_1 h_2)h_3 &= F(t_1 + t_2)h_3 = F((t_1 + t_2) + t_3) = F(t_1 + (t_2 + t_3)) \\ &= h_1 F(t_2 + t_3) = h_1(h_2 h_3)\end{aligned}$$

and finally if $F(t) = h$ then

$$F(t - t) = F(t_1)F(t)^{-1} = hh^{-1} = F(0) \tag{B.5}$$

and so $H \leq G$ as required. □

Appendix C

Quantum Mechanics And Hilbert Space

C.1 Properties of Hilbert Space

Definition C.1.1. Let $1 \leq p < \infty$ and X be a set under a counting measure on the sigma algebra $\mathcal{B}$ of the power set of X. Then we define $\ell^p(X)$ as

$$\ell^p(X) = L^p(X, \mathcal{B}, \mu) \tag{C.1}$$

Theorem C.1.2. *Let $\mathcal{H}$ be a Hilbert space and $\{e_i\}_{i\in I}$ be any orthonormal basis for $\mathcal{H}$. Then $\mathcal{H}$ is isometrically isomorphic to $\ell^2(I)$.*

Proof. Define the function $f : \mathcal{H} \to \ell^2(I)$ by

$$[f(x)](i) = \langle x, e_i \rangle \tag{C.2}$$

Using Parseval's identity, we have that

$$\|x\|_{\mathcal{H}}^2 = \sum_{i\in I} |\langle x, e_i \rangle|^2 = \sum_{i\in I} |[f(x)](i)|^2 = \|f(x)\|_{\ell^2(I)}^2 \tag{C.3}$$

so f is an isometry. Furthermore, it is a fundamental result that isometric functions on metric spaces are injective. Since all Hilbert spaces induce metric spaces, f must also be injective. Hence all that remains to be shown is that f is surjective.

Let $g \in \ell^2(I)$ so that

$$\sum_{i \in I} |g(i)|^2 < \infty \tag{C.4}$$

By the Riesz-Fisher theorem[10], $\sum_{i \in I} g(i) e_i \to h \in \mathcal{H}$ and

$$[f(h)](j) = \langle h, e_j \rangle = \sum_{i \in I} g(i) \langle e_i, e_j \rangle = g(j). \tag{C.5}$$

Since $g \in \ell^2(I)$ was arbitrary, we conclude the f is surjective. □

Theorem C.1.3. *Two Hilbert spaces H_1 and H_2 are isometrically isomorphic if and only if they have the same dimension.*

Proof. If the Hilbert spaces are isometrically isomorphic and $\{e_i\}_{i \in I}$ is a basis for H_1 then $\{[f(e_i)]\}_{i \in I}$ is a basis for H_2 and hence they have the same dimension, namely the cardinality of I.

Conversely, if two Hilbert spaces have the same dimension and one is indexed by I then the other can also be indexed by I. Hence both $H_1 \cong \ell^2(I)$ and $H_2 \cong \ell^2(I)$ so $H_1 \cong H_2$. □

Proposition C.1.4. *The Pauli matrices*

$$\sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{C.6}$$

form a basis for $i\mathfrak{su}(2)$. If we include the identity matrix

$$I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{C.7}$$

then $\{\sigma_X, \sigma_Y, \sigma_Z, I_2\}$ is a basis for $i\mathfrak{u}(2)$.

Proof. We first notice that $\mathcal{B}_1 = \{I_2, \sigma_X, \sigma_Y, \sigma_Z\}$ are all Hermitian, and that $\mathcal{B}_2 = \{\sigma_X, \sigma_Y, \sigma_Z\}$ are all traceless Hermitian. Since $i\mathfrak{su}(2)$ and $i\mathfrak{u}2$ are real vector spaces under addition, we can associate $M_2(\mathbb{C}) \cong \mathbb{C}^4$. In this case,

$$I_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad \sigma_X = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \qquad \sigma_Y = \begin{pmatrix} 0 \\ i \\ -i \\ 0 \end{pmatrix}, \sigma_Z = \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix} \tag{C.8}$$

Now these vectors are all clearly linearly independent and $\dim_{\mathbb{R}} \mathfrak{su}(2) = 3$, $\dim_{\mathbb{R}} \mathfrak{u}(2) = 4$. Thus $\mathcal{B}_1$ is a subset of four linearly independent elements in a four dimensional space and hence span $i\mathfrak{u}(2)$. Similarly for $\mathcal{B}_2$ in $\mathfrak{su}(2)$. □

Appendix D

Some Directions in Geometric Algorithms

In casting our problem into a real framework, we have omitted a great deal of the geometric interplay that is occurring within the function composition. Combined with the intractability of the method of Lagrange multipliers, it seems that the treatment necessitates a view on the evolution space itself. Herein the advantages and tools of differential geometry may prove useful, and our attempt to use such techniques to create an algorithm is what follows.

We offer some differential geometric techniques that begin to analyze the problem of driving one unitary operator to another. In particular, we take advantage of the Riemannian structure that can be defined on any smooth manifold and the sub-Riemannian structure of control problems in order to define a gradient. This gradient will give us the ability to analyze directions of steepest descent to find the desired control set.

D.1 Constructs on Riemmanian $\mathfrak{U}(d)$

Much of this work and theory was inspired by the papers [1, 2] whose endeavors are based in signal analysis. While there has been several works dedicated to solving optimization problems on manifolds [3, 75], the work of Abruden et al. was to explicitly analyze problems on unitary manifolds and to exploit the amenable properties of this structure. Abruden's goal was to minimize a real-valued objective function over $\mathfrak{U}(d)$. We hope to be able to adapt some of that author's techniques to create an algorithm for determining a control scheme.

Definition D.1.1. Let (M, g) be a Riemannian manifold and $f \in C^\infty(M)$. The *Riemannian gradient* of f is the vector field $\nabla f \in \Gamma(TM)$ such that $\forall X \in \Gamma(TM)$ we have

$$g(\nabla f, X) = df(X) = Xf. \tag{D.1}$$

We note that in the event that $(M, \mathcal{D}, g)$ is a *sub*-Riemannian manifold that the same definition holds for the *sub-Riemannian gradient* except that we add the restriction that (D.1) hold $\forall X \in \mathcal{D}$.

We need to analyze some of the particular constructs of the unitary manifold $\mathfrak{U}(d)$. In particular, for any given $X \in \mathfrak{U}(d)$ we would like to characterize the tangent space $T_X\mathfrak{U}(d)$ to which we have previously alluded, but yet to calculate. First we recall that if id is the identity element, it follows that $T_{\mathrm{id}}\mathfrak{U}(d) \cong \mathrm{Lie}(\mathfrak{U}(d)) \cong \mathfrak{u}(d)$. Let $\gamma(t) : [t_0, t_1] \to \mathfrak{U}(d)$ be a curve with $\gamma(t_0) = X$ and $\dot\gamma(t_0) = Y \in T_X\mathfrak{U}(d)$. Since $\gamma(t)$ is unitary for all time,

$$\gamma(t)^\dagger \gamma(t) = \mathrm{id}. \tag{D.2}$$

Differentiating (D.2) with respect to its time parameter and evaluating at $t = t_0$ we get

$$\begin{aligned} \frac{d}{dt}\bigg|_{t=0} \gamma(t)^\dagger\gamma(t) &= \dot\gamma(t)^\dagger\gamma(t) + \gamma(t)^\dagger\dot\gamma(t)\Big|_{t=0} && \text{(D.3a)} \\ &= Y^\dagger X + X^\dagger Y && \text{(D.3b)} \\ &= 0 && \text{(D.3c)} \end{aligned}$$

We notice that Y was chosen arbitrarily and uniquely defines each geodesic through X so that (D.3) yields a characterization of the tangent space $T_X\mathfrak{U}(d)$ as

$$T_X\mathfrak{U}(d) = \left\{ Y \in M_n(\mathbb{C}) \;\middle|\; Y^\dagger X + X^\dagger Y = 0. \right\} \tag{D.4}$$

Alternatively, the differential of the right translation action is an isomorphism [2] and so $(dR_{X^{-1}})_* : T_X\mathfrak{U}(d) \to \mathfrak{u}(d)$. This means that if $Y \in T_X\mathfrak{U}(d)$ then $(dR_{X^{-1}})_*(Y) \in \mathfrak{u}(d)$ and hence must be skew-Hermitian. From Lemma 2.2.45 and the fact that $X^{-1} = X^\dagger$, it follows that (YX^{-1}) is skew-Hermitian which means

$$(YX^\dagger) + (YX^\dagger)^\dagger = YX^\dagger + XY^\dagger = 0 \tag{D.5}$$

which is an equivalent characterization as that given by (D.4).

Next we will need to describe a *normal space*, which cannot be done in an embedding free way. In particular, we will need to consider $\mathfrak{U}(d)$ as a subspace of $M_{2n}(\mathbb{R})$. By

keeping the same inner-product, although extended to the ambient real-matrix space, we see that the orthogonal complement to $T_X\mathfrak{U}(d)$ is given by the left translation of Hermitian operators $i\mathfrak{u}(d)$ by X, or[2]

$$N_X\mathfrak{U}(d) = \left\{ XH \;\middle|\; H \in i\mathfrak{u}(d) \right\}. \tag{D.6}$$

With these constructs explicitly defined, we can move to work on calculating the Riemannian gradient. Let $f \in C^\infty(\mathfrak{U}(d))$ be a smooth, real-valued function. The complex matrix derivative is given by considering real and imaginary matrix derivatives as

$$\frac{\partial f}{\partial X^*}(X) = \frac{1}{2}\left(\frac{\partial f}{\partial X_R} + i\frac{\partial f}{\partial X_I} \right) \tag{D.7}$$

where X_R and X_I are the real and imaginary parts of X respectively. We can use this to define $G_X = \frac{\partial f}{\partial X^*}(X)$ which represents the derivative of f in the ambient space. By Definition D.1.1, the Riemannian gradient of f is given by $\nabla f \in \Gamma(T\mathfrak{U}(d))$ and satisfies

$$\langle \nabla f(X), Y_X \rangle_X = df_X(Y_X), \qquad \forall Y \in \Gamma(T\mathfrak{U}(d)). \tag{D.8}$$

The ambient inner-product is simply an extension of our intrinsic inner product, and so we demand that they must agree in general,

$$\langle \nabla f(X), Y_X \rangle_X = \langle G_X, X \rangle_{M_{2n}(\mathbb{R})}. \tag{D.9}$$

Viewing $\langle \cdot, \cdot \rangle_X$ as a restriction in the ambient space, we can rewrite (D.9) as

$$\left\langle \Gamma_X - \frac{1}{2}\nabla f, Y \right\rangle_{M_{2n}(\mathbb{R})} = 0, \qquad \forall Y \in T_X\mathfrak{U}(d). \tag{D.10}$$

Since this must hold for all points Y in the tangent space at X, $\Gamma_X - \frac{1}{2}\nabla f$ must inhabit the orthogonal complement to the tangent, or more precisely,

$$\Gamma_X - \frac{1}{2}\nabla f \in N_X\mathfrak{U}(d). \tag{D.11}$$

This implies that $\exists H \in i\mathfrak{u}(d)$ so that we can write $\Gamma_X - \frac{1}{2}\nabla f = XH$. We do not currently have a characterization of H, but can arrive at one by recognizing that since $\nabla f(X) \in T_X\mathfrak{U}(d)$ it must follow that

$$(\nabla f(X))^\dagger X + X^\dagger(\nabla f(X)) = 0. \tag{D.12}$$

By utilizing both (D.11) and (D.12) we can solve to find that $H = \frac{1}{2}\left[X^\dagger \Gamma_X + \Gamma_X^\dagger X\right]$. This yields the final expression for the Riemannian gradient as

$$\nabla f = \Gamma_X - X\Gamma_X^\dagger X. \tag{D.13}$$

A concern often considered in numerical analysis is to update parameters in a direction on which the objective function decreases. In single variate analysis, one can calculate whether the function is decreasing by calculating the sign of its derivative. Given an inner product in multiple dimensions, this is normally done by projecting a given direction onto the gradient and comparing sign of the projection. Negative signs corresponds to descent, positive signs to ascent, and zeros to orthogonal directions.

Definition D.1.2. Let M be a smooth manifold and $f \in C^\infty(M)$. Furthermore, take $X \in M$ and $\gamma : [t_0, t_1] \to M$ a curve in M with $\gamma(t_0) = X$ so that $\dot{\gamma}(0) \in T_X M$. Then $Y = \dot{\gamma}(0)$ is a *descent direction* for f at X if any of the following equivalent statements are true

$$df_X(Y) = \dot{\gamma}(0)f = \frac{d}{dt} f(\gamma(t))\bigg|_{t=0} < 0. \tag{D.14}$$

These techniques can be applied to control problems if properly cast into a numerical optimization framework. Though such attempts have met with limited success, we present some of our ideas and methods in the following section.

D.2 Application to Control Methods

In section D.1 we analyzed the work of Abruden et al. in creating steepest-descent optimization routines using the Riemannian gradient. Here we will discuss some of the future work that will be done using the ideas developed in [1, 2]. We note that additional theoretical results need to be established before one can properly discuss issues such as stability and convergence of these algorithms. While we have implemented many of these procedures, we have met with only very limited success, discussed below on page 133. Nonetheless, they will be mentioned here due to their ability to potentially increase the speed with which these algorithms can be run.

First, we choose to make use of the Riemannian gradient given by (D.13). Consider the normalized operator form of the Schrödinger equation given by (4.1) and assume that

$H(u)$ takes the form of an affine-control system with drift.

$$\frac{d}{dt}X(t) = i\left(H_0 + \sum_{i=1}^{m} u_i(t)H_i\right)X(t), \qquad X(0) = \mathrm{id}. \tag{D.15}$$

The surjectivity of the exponentiation operator guarantees that given any unitary operator $X_d \in \mathfrak{U}(d)$, there exists $H_d \in i\mathfrak{u}(d)$ such that $X_d = \exp[-iH_d]$. However, the skew-Hermitian operator $iH_d \in \mathfrak{u}(d)$ will generally not be among the set of admissible controls fields, preventing us from blindly applying this technique. Instead, by using a steepest descent algorithm, we can iteratively create a path of local geodesics that drive any initial state X_0 to the final state X_d. Consider the following algorithm given by Abruden et al. [1].

Given an initial state $X_0 \in \mathfrak{U}(d)$ and an objective function $f \in C^\infty(\mathfrak{U}(d))$ to be minimized, consider the following algorithm

1. Initialize $k = 0$ and $Y_k = X_0$.
2. Compute the gradient of the cost function on Euclidean space $\Gamma_k = \frac{\partial f}{\partial X^*}(Y_k)$
3. Compute the gradient direction on the Riemannian space $G_k = \Gamma_k Y_k^\dagger - Y_k \Gamma_k^\dagger$
4. Determine a value $\mu > 0$ such that $P_k = \exp(-\mu G_k)$ is a descent direction.
5. Update $Y_{k+1} = P_k Y_k$ and $k = k + 1$. Check a stopping criterion, otherwise re-iterate steps 2 through 5.

In our case, the objective function is just $f(X) = d(X, X_d)^2$. Applying this algorithm to this problem seems like a trivial exercise, as we know *a priori* the objective function is minimized by X_d. However, the point is not that we are trying to find the element X that minimizes f, but rather that the algorithm gives a construction of X_d in the form

$$X_d = \exp[-\mu_k G_k]\exp[-\mu_{k-1}G_{k-1}] \times \cdots \times \exp[-\mu_1 G_1]. \tag{D.16}$$

In the limiting regime of bang-bang methods, a set of m controls yields only 2^m possible control configurations. One can see that this is true by considering the number of m-multisets of the logical on/off set $\{0, 1\}$, for which simple combinatorial arguments yield that there are 2^m possibilities. Equivalently, if $\{H_i\}_{i=1}^m$ are the control fields, the set of all possible control configurations is the powerset of $\{H_i\}_{i=1}^m$ which has order 2^m. Since

this is a distinctly finite subset of a continuum, it naturally has zero measure and hence is a null set amongst the set of all unitaries. Similarly, in the non bang-bang regime our distribution of admissible controls often has non-zero codimension and consequently will span a strict linear subspace of the tangent bundle $T\mathfrak{U}(d)$.

This implies that there may still be an issues with the G_i of (D.16) not representing admissible controls. However, we have now turned our problem from a global one into a local one, on which we may apply local optimization techniques. In order to exploit this algorithm, one can examine the set of feasible controls to find which yields a direction of greatest descent, or we can try to approximate the path given by the geodesics constructed in the algorithm.

In the first approach, we analyze the control sets that give the direction of greatest descent. We note that if X_0 is a given point and $H_{X_0} \in T_{X_0}\mathfrak{U}(d)$ is an element of the tangent space, the geodesic emanating from X_0 in the "direction" of H_{X_0} is $\gamma(t) = \exp\left[-tH_{X_0}\right] X_0$. Hence $\dot{\gamma}(0) = H_{X_0}$ is a descent direction if and only if (D.14) is satisfied, namely

$$\dot{\gamma}(0)f = \frac{d}{dt} f(\gamma(t))\bigg|_{t=0} < 0. \tag{D.17}$$

Two problems present themselves here. The first is that there may be a high-frequency control that is not initially a descent direction but may yield a faster, higher-fidelity result. This can be overcome by analyzing the high-end and low-end frequencies of each control; a method which is discussed in [2] for almost-periodic elements of $\mathfrak{U}(d)$. One can then make an appropriate decision as to the optimal control at that point. Secondly, this method is very prone to finding local minima of the function f. In fact, the identity matrix is itself a local minima of f on the manifold, requiring that we actually use a perturbed or randomized initial unitary X_0 rather than the identity.

Implementations of this algorithm are at first very promising. We note that by using the gradient calculation on the manifold we can remove many of the matrix exponential calculations that appear in Kaya-Huneault, Real-Embedding, and GRAPE. We can improve this further by realizing we can remove some of the matrix exponentials used in the line search in the direction of the gradient. By exploiting the fact that $\exp\left(\frac{t}{m}A\right)^m = \exp tA$, we can do an m-fold interpolation-based line search so that only one matrix exponential needs to be calculated. Unfortunately, these steepest descent methods find local minima frequently, meaning that convergence poses a serious issue.

Alternatively, we can completely avoid the computational algorithm and instead focus on analyzing steepest descent directions using the provided gradient. As mentioned previously, the assumption that our control law occupies the bang-bang regime dictates

a countable set of control laws. More precisely, given an m-input system there are 2^m possible control states. One might choose to individually examine the magnitude of each control state the corresponds to a descent direction, implementing the one that drives us closest to the objective and re-iterating. However, there is a much better alternative to this method. Instead of finding a Riemannian gradient and trying to find the control for which the best approximation is provided, we can instead calculate the sub-Riemannian gradient which will immediately give us the controls that yield a descent direction.

D.3 Constructs on Sub-Riemannian $\mathfrak{U}(d)$

We now provide a calculation as to the sub-Riemannian gradient. Let $f : \mathfrak{U}(d) \to \mathbb{R}$ be an objective function. We first recall the definition of a sub-Riemannian gradient as commented upon in Definition D.1.1.

Definition D.3.1. Let $(M, \mathcal{D}, g)$ be a sub-Riemannian manifold and $f \in C^\infty(M)$. The *sub-Riemannian gradient* of f is the vector field $\nabla_h f \in \Gamma(TM)$ such that $\forall X \in \mathcal{D}$ we have

$$g(\nabla_h f, X) = df(X) = Xf. \tag{D.18}$$

We use here the notation ∇_h instead of ∇ to make clear the sub-Riemannian contribution. The h stands for "horizontal."

By using an orthonormal frame for the distribution, we can give a more computationally friendly version of the sub-Riemannian gradient.

Proposition D.3.2 (Calin [20])**.** *Let $f \in C^\infty(M)$ and $\{E_i, \ldots, E_k\}$ be an orthonormal frame at $p \in M$. Then*

$$(\nabla_h f)_p = \sum_{i=1}^{k} (E_i)_p f \ (E_i)_p. \tag{D.19}$$

Proof. By definition of a frame, the $\{E_1, \ldots, E_k\}$ are linearly independent in T_qM for all q in a sufficiently small neighbourhood of p. This implies that we can write the horizontal gradient at T_qM in terms of $(E_i)_q$ as

$$(\nabla_h f)_q = \sum_{i=1}^{k} c^i(q)(E_i)_q \tag{D.20}$$

for some set of constants $c^i(q)$ that are yet to be determined. By applying the metric on (D.20) and some E_j we get

$$\begin{aligned} g_q((\nabla_h f)_q, (E_j)_q) &= \sum_{i=1}^{k} c^i(q)\ g\left((E_i)_q, (E_j)_q\right) \\ &= \sum_{i=1}^{k} c^i(q)\delta_{ij} \\ &= c^j(q) \end{aligned}$$

By the definition of the sub-Riemannian metric, this gives

$$c^j(q) = g_q((\nabla_h f)_q, (E_j)_q) \overset{(D.18)}{=} (E_j)_q f. \tag{D.21}$$

□

Given the m-input, affine Schrödinger equation

$$\frac{d}{dt}X(t) = i\left(H_0 + \sum_{i=1}^{m} u_i(t)H_i\right)X(t), \qquad X(0) = \mathrm{id}. \tag{D.22}$$

let us make the assumption that our vector fields $(H_1X, \ldots, H_mX)$ are an orthonormal frame in a neighbourhood of X. In order to compute the sub-Riemannian gradient at X, it is sufficient to compute $(H_iX)f$ for each control field H_i.

Proposition D.3.3 (Lee [52])**.** *Suppose M is a smooth manifold, $\gamma : J \to M$ is a smooth curve and $f : M \to \mathbb{R}$ is a smooth function. Then the derivative of the real-valued function $f \circ \gamma : \mathbb{R} \to \mathbb{R}$ is given by*

$$(f \circ \gamma)'(t) = df_{\gamma(t)}(\gamma'(t)). \tag{D.23}$$

Proof. This proof follows directly from the definition of all involved terms. Indeed for any fixed $t_0 \in J$ we have

$$\begin{aligned} df_{\gamma(t_0)}(\gamma'(t_0)) &= \gamma'(t_0)f \\ &= \left(\gamma_*\left.\frac{d}{dt}\right|_{t=t_0}\right) f \\ &= \left.\frac{d}{dt}\right|_{t=t_0} (f \circ \gamma) \\ &= (f \circ \gamma)'(t_0). \end{aligned}$$

□

Define $\gamma(t) = e^{tH_i}X$ and notice that $\gamma(0) = X$, $\gamma'(0) = H_iX$. By applying Proposition D.3.3 we have that

$$\left.\frac{d}{dt}\right|_{t=0} f\left(e^{tH_i}X_i\right) = d_X f(H_iX) = (H_iX)f. \tag{D.24}$$

For our purposes, we recall that if $X_d \in \mathfrak{U}(d)$ is the desired objective unitary operator, we defined our function $f(X)$ as

$$f(X) = \langle X - X_d, X - X_d\rangle = 2d - 2\Re\langle X, X_d\rangle. \tag{D.25}$$

This allows us to state the following:

Lemma D.3.4. *Given that $f(X)$ is defined as in* (D.25) *it follows that*

$$\left.\frac{d}{dt}\right|_{t=0} f\left(e^{tH_i}X_i\right) = -2\Re\langle H_iX, X_d\rangle. \tag{D.26}$$

Proof. This is an exercise in applying definitions. Recall that by definition of the exponential we have

$$e^{tH_i}X = \left(\sum_{k=0}^{\infty}(tH_i)^k\right)X = X + tH_iX + \frac{t^2}{2}H_i^2X + \cdots \tag{D.27}$$

Substituting (D.27) into $f(X)$ yields

$$f(e^{tH_i}X) = 2d - 2\Re\left\langle e^{tH_i}X, X_d\right\rangle = 2d - 2\Re\sum_{k=0}^{\infty} t^k\left\langle H_i^kX, X_d\right\rangle. \tag{D.28}$$

The act of differentiating causes the constant term $2d$ and the term corresponding to $k = 0$ to disappear. Afterwards, setting $t = 0$ causes all terms corresponding to $k > 1$ disappear. Consequently, we are left with only the $k = 1$ term which implies that

$$\left.\frac{d}{dt}\right|_{t=0} f\left(e^{tH_i}X_i\right) = -2\Re\langle H_iX, X_d\rangle$$

which is precisely what we required. □

Theorem D.3.5. *The sub-Riemannian gradient of the objective function $f(X)$ as defined* (D.25) *corresponds to the control of steepest descent in an unbounded, driftless system.*

Proof. Applying Lemma D.3.4 to the definition of the sub-Riemannian gradient, we get that the gradient at $X \in \mathfrak{U}(d)$ is given by

$$(\nabla_h f)_X = \sum_{i=1}^{m} \Big[-2\Re \langle X, X_d \rangle \Big] H_i X. \tag{D.29}$$

Comparing to the driftless, affine, Schrödinger equation

$$i\frac{d}{dt}X(t) = \sum_{i=1}^{m} u_i(t) H_i X \tag{D.30}$$

we see that by setting $u_i(t) = 2\Re \langle H_i X, X_d \rangle$ we get

$$i\frac{d}{dt}X(t) = -(\nabla_h f)_X, \tag{D.31}$$

the differential equation corresponding to steepest-descent. □

We would like to make two comments on this theorem. The first is that we assumed that $\{H_i X\}_{i=1}^m$ is orthonormal. In the event that they are not orthonormal, we can apply a Gram-Schmidt orthonormalization procedure to find a sufficient basis for a fixed tangent space. This will affect the statement of Theorem D.3.5, but will not change the fact that the control amplitudes can be found from the sub-Riemannian gradient. The second statement is that the previous theorem requires that our system be driftless. Whether the case of drift systems can be accommodated into this framework has yet to be determined and will be analyzed in future work.

References

[1] T. Abrudan, J. Eriksson, and V. Koivunen. Steepest descent algorithms for optimization under unitary matrix constraint. *IEEE Transactions on Signal Processing*, 56(3):1134 –1147, March 2008. 128, 131, 132

[2] T. Abrudan, J. Eriksson, and V. Koivunen. Conjugate gradient algorithm for optimization under unitary matrix constraint. *Signal Processing*, 89(9):1704–1714, SEP 2009. 128, 129, 130, 131, 133

[3] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008. 128

[4] A. Agrachev and T. Chambrion. An estimation of the controllability time for single-input systems on compact lie groups. *ESAIM: COCV*, 12(3):409–441, 2006. 45, 61

[5] A. Agrachev and Y. Sachkov. *Control Theory from a Geometric Viewpoint*. Springer, 2004. 30, 41, 100, 110

[6] C. Altafini. Controllability of quantum mechanical systems by root space decomposition of $\mathfrak{su}(n)$. *Journal of Mathematical Physics*, 43(5):2051–2062, 2002. 45

[7] M. Athans. The matrix minimum principle. *Information and Control*, 11(5-6):592 – 606, 1967. 8, 9

[8] A. Baker. *Matrix Groups: An Introduction to Lie Group Theory*. Springer Undergraduate Mathematics Series. Springer, 2002. 18, 39, 72

[9] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, 52(5):3457–3467, Nov 1995. 83

[10] R. Beals. *Analysis: An Introduction*. Cambridge University Press, 2004. 127

[11] B. Bonnard, J.-B. Caillau, and E. Trélat. Second order optimality conditions in the smooth case and applications in optimal control. *ESAIM: COCV*, 13(2):207–236, 2007. 100

[12] B. Bonnard and M. Chyba. *Singular Trajectories and their Role in Control Theory*. Springer, 2003. 29, 108

[13] B. Bonnard, M. Chyba, and D. Sugny. Time-minimal control of dissipative two-level quantum systems: The generic case. *IEEE Transactions on Automatic Control*, 54:2598–2610, 2009. 62, 108

[14] B. Bonnard and I. Kupka. Generic properties of singular extremals. *Annales de l'Institut Henri Poincare*, 14(2):167–186, 1997. 113

[15] B. Bonnard and D. Sugny. Time-minimal control of dissipative two-level quantum systems: The integrable case. *SIAM Journal on Control and Optimization*, 48(3):1289–1308, 2009. 62, 108

[16] 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. 61, 96

[17] R. W. Brockett. System theory on group manifolds and coset spaces. *SIAM Journal on Control*, 10(2):265–284, 1972. 57

[18] R. L. Burden and J. D. Faires. *Numerical Analysis*. Thomson Brooks/Cole, 2005. 64

[19] K. Burns and M. Gidea. *Differential Geoemtry and Topology With a View to Dynamical Systems*. Chapman and Hall, 2005. 9, 10, 13, 16

[20] O. Calin and D.-C. Chang. *Sub-Riemannian Geometry*. Cambridge University Press, 2009. 24, 134

[21] D. E. Chang. A simple proof of the pontryagin maximum principle on manifolds. *Automatica*, 47(3):630 – 633, 2011. 33

[22] Y. Chitour, F. Jean, and E. Trélat. Genericity results for singular trajectories. *Journal of Differential Geometry*, 73:45–73, 2006. 13, 106

[23] J. B. Conway. *A Course In Operator Theory*. American Mathematical Society, 1999. 75

[24] D. D'Alessandro. Uniform finite generation of compact lie groups. *Systems & Control Letters*, 47(1):87–90, SEP 16 2002. 51

[25] D. D'Alessandro. General methods to control right-invariant systems on compact lie groups and multilevel quantum systems. *Journal of Physics A - Mathematical and Theoretical*, 42(39), OCT 2 2009. 51, 53

[26] M. R. Dowling and M. A. Nielsen. The geometry of quantum computation. `arXiv:quant-ph/0701004v1`, Dec. 2007. 48

[27] H. N. S. Earp and J. K. Pachos. A constructive algorithm for the cartan decomposition of $\mathfrak{SU}(2^n)$. *Journal of Mathematical Physics*, 46(8):082108, 2005. 61

[28] K. Erdmann and M. J. Wildon. *Introduction To Lie Algebras*. Springer Undergraduate Mathematics Series. Springer, 2006. 19

[29] E. Fredkin and T. Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21:219–253, 1982. 10.1007/BF01857727. 83

[30] W. Fulton and J. Harris. *Representation Theory: A First Course*. Springer-Verlag, 1991. 21

[31] R. Goodman and N. R. Wallach. *Representations and Invariants of the Classical Groups*. Cambridge University Press, 1998. 121

[32] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, STOC '96, pages 212–219, New York, NY, USA, 1996. ACM. 1

[33] T.-S. Ho, J. Dominy, and H. Rabitz. Landscape of unitary transformations in controlled quantum dynamics. *Phys. Rev. A*, 79(1):013422, Jan 2009. 48, 50

[34] M. Hsieh and H. Rabitz. Optimal control landscape for the generation of unitary transformations. *Phys. Rev. A*, 77(4):042306, Apr 2008. 48, 49

[35] M. Hsieh, R. Wu, H. Rabitz, and D. Lidar. Optimal control landscape for the generation of unitary transformations with constrained dynamics. *Phys. Rev. A*, 81(6):062352, Jun 2010. 48, 50

[36] L. Hsu. Calculus of variations via the griffiths formalism. *Journal of Differential Geometry*, 36(3):551–589, 1992. 106

[37] R. Huneault. Time-optimal control of closed quantum systems. Master's thesis, University of Waterloo, 2009. 3, 8, 9, 41, 42, 65, 67, 68, 71, 73, 117

[38] B. D. Johnson. The nonholonomy of the rolling sphere. *The American Mathematical Monthly*, 114(6):pp. 500–508, 2007. 25

[39] V. Jurdjevic. *Geometric Control Theory*. Cambridge University Press, 1997. 4, 13, 22, 23, 32, 33, 124

[40] V. Jurdjevic and H. J. Sussmann. Control systems on lie groups. *Journal of Differential Equations*, 12(2):313 – 329, 1972. 44, 51

[41] C. Y. Kaya, S. K. Lucas, and S. T. Simakov. Computations for bang-bang constrained optimal control using a mathematical programming formulation. *Optimal Control Applications and Methods*, 25(6):295–308, 2004. 67, 73

[42] C. Y. Kaya and J. L. Noakes. Computations and time-optimal controls. *Optimal Control Applications and Methods*, 17(3):171–185, 1996. 67

[43] C. Y. Kaya and J. L. Noakes. Computational method for time-optimal switching control. *Journal of Optimization Theory and Applications*, 117(1):69–92, 2003. 67

[44] P. Kaye, R. Laflamme, and M. Mosca. *An Introduction to Quantum Computing*. Oxford University Press, 2006. 83

[45] N. Khaneja, R. Brockett, and S. J. Glaser. Time optimal control in spin systems. *Phys. Rev. A*, 63(3):032308, Feb 2001. 2, 57, 59, 60

[46] N. Khaneja and S. J. Glaser. Cartan decomposition of su(2(n)) and control of spin systems. *Chemical Physics*, 267(1-3):11–23, JUN 1 2001. 60, 62

[47] 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. 57, 59

[48] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrggen, 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, 2005. 3, 54, 56, 64

[49] E. Kreyszig. *Introductory Functional Analysis with Applications*. John Wiley and Sons, 1978. 35

[50] A. B. Kurzhanski. Pontryagin maximum principle. Springer Online Encyclopedia. 9

[51] M. Lapert, Y. Zhang, M. Braun, S. J. Glaser, and D. Sugny. Singular extremals for the time-optimal control of dissipative spin $\frac{1}{2}$ particles. *Phys. Rev. Lett.*, 104(8):083001, Feb 2010. 62, 108

[52] J. M. Lee. *Introduction to Smooth Manifolds*. Springer, 2002. 12, 13, 17, 19, 20, 25, 26, 33, 52, 121, 135

[53] C. Moler and C. V. Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):3–49, 2003. 75

[54] R. Montgomery. Singular extremals on lie groups. *Mathematics of Control, Signals, and Systems (MCSS)*, 7:217–234, 1994. 10.1007/BF01212270. 99

[55] R. Montgomery. A survey of singular curves in sub-riemannian geometry. *Journal of Dynamical and Control Systems*, 1:49–90, 1995. 10.1007/BF02254656. 24

[56] R. Montgomery. *A Tour of Subriemannian Geometries, Their Geodesics and Applications*. American Mathematical Society, 2002. 100, 103, 106

[57] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 1 edition, October 2000. 83

[58] M. A. Nielsen, M. R. Dowling, M. Gu, and A. C. Doherty. Optimal control, geometry, and quantum computing. *Phys. Rev. A*, 73(6):062323, Jun 2006. 47

[59] M. A. Nielsen, M. R. Dowling, M. Gu, and A. C. Doherty. Quantum computation as geometry. *Science*, 311(5764):1133–1135, 2006. 47, 48

[60] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999. 79, 80, 81

[61] L. S. Pontryagin, V. G. Boltyanskii, R. Gamkrelidze, and E. F. Mishchenko. *The Mathematical Theory of Optimal Processes*. Wiley Interscience, 1962. 4, 6, 106

[62] H. Rabitz, T.-S. Ho, M. Hsieh, R. Kosut, and M. Demiralp. Topology of optimally controlled quantum mechanical transition probability landscapes. *Phys. Rev. A*, 74(1):012721, Jul 2006. 48, 50

[63] H. Rabitz, M. Hsieh, and C. Rosenthal. Landscape for optimal control of quantum-mechanical unitary transformations. *Phys. Rev. A*, 72(5):052337, Nov 2005. 48, 49, 50

[64] H. A. Rabitz, M. M. Hsieh, and C. M. Rosenthal. Quantum optimally controlled transition landscapes. *Science*, 303(5666):1998–2001, 2004. 48, 50

[65] J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics of Computation*, 145:367–393, 1990. 63

[66] S. Sastry. *Nonlinear Systems: Analysis, Stability, and Control*. Springer, 1999. 19, 21

[67] S. G. Schirmer, H. Fu, and A. I. Solomon. Complete controllability of quantum systems. *Phys. Rev. A*, 63(6):063410, May 2001. 45

[68] M. R. Sepanski. *Algebras*. American Mathematical Society, 2010. 49

[69] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):pp. 303–332, 1999. 1

[70] E. D. Sontag. *Mathematical Control Theory*. Springer, 2nd edition, 1998. 4

[71] D. Sugny, C. Kontz, and H. R. Jauslin. Time-optimal control of a two-level dissipative quantum system. *Phys. Rev. A*, 76(2):023419, Aug 2007. 108

[72] H. J. Sussman and G. Tang. Shortest paths for the reeds-shepp car. a worked out example of the use of geometric techniques in nonlinear optimal control. Technical report, SYCON - Rutgers Center for Systems and Control, September 1991. 63

[73] R. Wu, J. Dominy, T.-S. Ho, and H. Rabitz. Singularities of quantum control landscapes. `arXiv:0907.2354v1 [quant-ph]`, July 2009. 108

[74] R. Wu, H. Rabitz, and M. Hsieh. Characterization of the critical submanifolds in quantum ensemble control landscapes. *Journal of Physics A: Mathematical and Theoretical*, 41(1):015006, 2008. 48, 50

[75] Y.Yang. Globally convergent optimization algorithms on riemannian manifolds: Uniform framework for unconstrained and constrained optimization. *Journal of Optimization Theory and Applications*, 132(2):245–265, February 2007. 128

[76] R. Zeier, H. Yuan, and N. Khaneja. Time-optimal synthesis of unitary transformations in a coupled fast and slow qubit system. *Phys. Rev. A*, 77(3):032332, Mar 2008. 62