A Generalized Label Correcting Method for Optimal Kinodynamic Motion Planning

by

Brian A. Paden

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2017

Author . . . .

Signature redacted

Department of Mechanical Engineering

June 5, 2017

Certified by . . . .

Signature redacted

Emilio Frazzoli
Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by . .

Signature redacted

Domitilla Del Vecchio
Associate Professor of Mechanical Engineering
Chair, Thesis Committee

Certified by .

Signature redacted

Karl Iagnemma
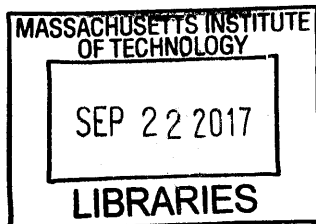Principal Research Scientist
Member, Thesis Committee

Accepted by . . . .

Signature redacted

Rohan Abeyaratne
Chair, Department Committee on Graduate Theses

MITLibraries

DISCLAIMER NOTICE

Due to the condition of the original material, there are unavoidable flaws in this reproduction. We have made every effort possible to provide you with the best copy available.

Thank you.

The images contained in this document are of the best quality available.

A Generalized Label Correcting Method for Optimal Kinodynamic Motion Planning

by

Brian A. Paden

Submitted to the Department of Mechanical Engineering
on June 5, 2017, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Nearly all autonomous robotic systems use some form of motion planning to compute reference motions through their environment. An increasing use of autonomous robots in a broad range of applications creates a need for efficient, general purpose motion planning algorithms that are applicable in any of these new application domains.

This thesis presents a resolution complete optimal kinodynamic motion planning algorithm based on a direct forward search of the set of admissible input signals to a dynamical model. The advantage of this generalized label correcting method is that it does not require a local planning subroutine as in the case of related methods.

Preliminary material focuses on new topological properties of the canonical problem formulation that are used to show continuity of the performance objective. These observations are used to derive a generalization of Bellman's principle of optimality in the context of kinodynamic motion planning. A generalized label correcting algorithm is then proposed which leverages these results to prune candidate input signals from the search when their cost is greater than related signals.

The second part of this thesis addresses admissible heuristics for kinodynamic motion planning. An admissibility condition is derived that can be used to verify the admissibility of candidate heuristics for a particular problem. This condition also characterizes a convex set of admissible heuristics. A linear program is formulated to obtain a heuristic which is as close to the optimal cost-to-go as possible while remaining admissible. This optimization is justified by showing its solution coincides with the solution to the Hamilton-Jacobi-Bellman equation. Lastly, a sum-of-squares relaxation of this infinite-dimensional linear program is proposed for obtaining provably admissible approximate solutions.

Thesis Supervisor: Emilio Frazzoli
Title: Professor of Aeronautics and Astronautics

Acknowledgments

I would like to express my gratitude to my advisor Professor Emilio Frazzoli for his input and support through my graduate studies. Emilio has given me tremendous freedom in pursuing the research questions I found most interesting and provided skillful guidance to keep me moving in the right direction. I have also enjoyed numerous opportunities that were a direct consequence of my affiliation with Emilio's research group. Among these were several exciting internships working on autonomous systems and the opportunity to spend a year doing research at ETH.

I am also grateful to my committee members, Professor Domitilla Del Vecchio and Karl Iagnemma for their valuable feedback and advice in out meetings over the past several years.

Additionally, I would like to thank my colleagues and friends with whom I have worked over the past four years. Sze Zheng Yong was a great mentor in my first year at MIT and we were able to collaborate on a number of interesting projects. The afternoon chalkboard sessions with Dmitry Yershov were stimulating and memorable. They contributed significantly to the ideas in the first part of this thesis. The second part of this thesis related to heuristics for kinodynamic planning problems came from a collaboration with Valerio Varricchio on a course project in Professor Pablo Parrilo's algebraic techniques class.

For Stephanie

Contents

Chapter 1

Introduction

Robotics and automation have been reshaping the world economy with widespread use in industries such as manufacturing, agriculture, transportation, defense, and medicine. Advances to the theoretical foundations of the subject together with a seemingly endless stream of new sensing and computing technology is making autonomous robots capable of increasingly complex tasks. One of the emerging new applications is driverless vehicles [PvY$^+$16] and transportation systems which have the potential to eliminate urban congestion [STZ$^+$14] and dramatically reduce the roughly 1.24 million lives lost each year in road traffic accidents [VPO13].

The sophistication and complexity of modern robotic systems requires entire fields of research devoted to individual subsystems. These can be divided into two major categories. The first, sensing and perception which inform the system about its state and the surrounding environment; the second, planning and control which makes use of sensor data, processed by the perception system, and selects appropriate actions to accomplish task specifications.

While both of these are subjects of active research, the scope of this thesis falls into the latter category of planning and control. Specifically, it focuses on the planning of motions through an environment that satisfy a dynamical model of the system while optimizing a measure of performance.

1.1 Background and Motivation

Research on planning and control for robotic systems is addressed with a combination of techniques from theoretical computer science and control theory. High level decision making and task planning problems generally involve making selections from a discrete set of alternatives. The techniques from theoretical computer science lend themselves well to these problems given their discrete nature. Central research questions involve: determining expressive problem formulations that can be utilized in a large variety of applications, determining the computational complexity of a particular problem formulation, and then designing sound and correct algorithms which realize this complexity. At the other end of the spectrum, a feedback control system faithfully executes reference motions given a potentially complex dynamical model of

the system. Central research questions in control theory are the stability of the system in the presence of feedback control, and robustness to disturbances and modeling errors.

While computer science provides the tools for high level planning and control theory provides the tools to execute a reference motion, the actual planning of a robot's reference motion falls in the middle ground between these two disciplines and has been treated almost independently by the two. The main specifications for the planned motion are: (i) the motion must originate from a prespecified initial state and terminate in a set of goal states; (ii) it must satisfy constraints on the state, such as obstacle avoidance, along the planned motion; (iii) there must exist a control signal which, together with the planned motion, satisfies the differential equation modeling the system; (iv) lastly, a measure of cost must be minimized.

1.1.1 Methods from Optimal Control

A number of techniques for solving this problem have been developed within the optimal control literature where greater emphasis has been placed on dealing with differential constraints and less on state constraints. The classical approach is to use extensions of the calculus of variations to express necessary first order optimality conditions of a solution. One celebrated approach is known as Pontryagin's minimum principle after Lev Pontryagin who pioneered the extension to optimal control [PVGBM62]. The first order optimality conditions are a number of differential equations that must satisfy initial and terminal state constraints, also known as a two-point boundary value problem. The two-point boundary value problem can be solved in closed form in a number of examples, but in general it requires a numerical method to determine the appropriate boundary values. The shooting method is the standard numerical method for solving the two-point boundary value problem. However, extreme sensitivity of the terminal boundary value with respect to the initial boundary value makes it difficult to solve in most examples [BH75, pg. 214].

Direct methods have become a favored approach. These methods approximate the trajectory and control as a finite-dimensional vector space and then directly optimize the performance objective with a nonlinear programming algorithm. Some of the popular approaches are the direct shooting method, reviewed in [Bet98], which is particularly easy to implement; orthogonal collocation methods [BHTR06] which offer a good numerical approximation with a relatively small number of basis vectors; and direct collocation methods [HP87] which provide an approximation resulting in a sparse Hessian matrix for the performance objective.

The direct optimization methods rapidly converge to a locally optimal solution when they are given a suitable initial guess. However, constructing an initial guess is more of an art than a science and if the initial guess does not satisfy the constraints, many solvers may fail to find a feasible solution at all. Additionally, direct optimization methods will converge to a locally optimal solution which may be unacceptably different than any globally optimal solutions. These issues are exacerbated in complex environments that can introduce numerous local minima in the performance objective. Thus, direct optimization methods are not sound in the sense that they may fail

to find a solution when one exists. In real-time motion planning applications where the output of the motion planner can be safety critical, these methods need to be used with a great deal of caution.

1.1.2   Methods from Computational Geometry

A classic problem in computational geometry is the mover's problem (also called the piano or couch mover's problem) which had received attention from the computational geometry community during the 1970s. A generalization of the mover's problem was gaining interest with the increasing use of industrial manipulators in manufacturing in the late 1970s. The generalized mover's problem consider's planning a motion for multiple polyhedra freely linked at distinguished vertices (to model an industrial manipulator arm).

In a famous paper by Reif [Rei79], an algorithm is provided for the mover's problem whose complexity is polynomial in the number of constraints defined by the obstacles. Further, Rcif shows that the generalized mover's problem is PSPACE-hard with respect to the number of degrees of freedom meaning that the difficulty of solving motion planning problems grows rapidly as the degrees of freedom increase. Several years later Schwartz and Sharir presented a cell-decomposition algorithm [SS83] which solved the generalized mover's problem for algebraic (instead of polyhedral) bodies with complexity in $O((2n)^{3^{r+1}} \cdot m^{2^r})$ where r is proportional to the number of degrees of freedom of the robot, n is proportional to the algebraic degree of the constraints, and m is the number of polynomials describing the constraints. The cell decomposition algorithm is most often cited for the doubly exponential complexity in the degrees of freedom of the robot which is not so disappointing considering Reif's results several years earlier. However, the important observation is that the complexity is polynomial with respect to the number of obstacles in the environment. Canny later provided an algorithm that solved the generalized mover's problem with complexity which is only exponential in the degrees of freedom [Can87].

In contrast to the trajectory optimization problems addressed by the optimal control community, the classical kinematic motion planning problems are only concerned with obstacle avoidance. For robot manipulators there is a straight-forward justification for neglecting the robot dynamics. The dynamics of robot manipulators determined by the principles of classical mechanics [AM78] can almost universally be written as

$$M(q(t), \dot{q}(t))\ddot{q}(t) + C(q(t), \dot{q}(t))\dot{q}(t) + G(q(t)) = B(q(t))\tau(t), \qquad (1.1.1)$$

where $q(t)$ is the vector of generalized coordinates for the robot's configuration and $\tau(t)$ is the vector of generalized control forces applied by the actuators. When there is at least one actuator for each generalized coordinate and the matrix B has rank equal to the number of generalized coordinates for every configuration, any twice-differentiable time parameterization of a planned motion q can be executed with the control forces $\tau(t)$ solving the manipulator equation (1.1.1). When these conditions are met, the robot is said to be fully actuated which is typically the case for robot

manipulators.

1.1.3 Approximate Methods for Motion Planning

The growing robotics industry in the late 1980s and 1990s called for practical solutions to the motion planning problem. With the disappointing complexity of available complete algorithms, researchers began developing practical techniques without theoretical guarantees, that worked well in practice [Kha86, HA92, RK92].

Another practical approach with some theoretical justification was to seek methods with *resolution completeness* [BLP85], meaning that a the output of the algorithm is correct for some (not known a-priori) sufficiently high resolution. The analogue for randomized algorithms is *probabilistic completeness* [BKM$^+$96]. The concept of probabilistic completeness was introduced at around the same time as the probabilistic roadmap (PRM) algorithm [KSLO96], whose effectiveness triggered a paradigm shift in motion planning research towards sampling-based approximate methods. One of the more attractive features of the algorithm is that the probability of failing to correctly determine the feasibility of a motion planning problem converges to 0 at an exponential rate in the number of random samples. The basic principle of the PRM algorithm is to randomly sample a large number admissible robot configurations, and then construct a graph by connecting nearby configurations with an edge if the line between them does not contain inadmissible configurations.

With the PRM algorithm effectively solving classical motion planning problems, attention turned to planning for systems where dynamical constraints cannot be neglected or *kinodynamic motion planning*. The difficulty in applying the PRM to kinodynamic motion planning is that the construction of edges by linear interpolation between configurations may not be a dynamically feasible motion. A simple adaptation is to replace linear interpolation by a local planning or steering subroutine, but this complicates individual implementations and places a burden on the user of the algorithm to provide this subroutine which itself must solve a kinodynamic motion planning problem. The first algorithms addressing the kinodynamic motion planning problem without a steering subroutine were the expansive space trees (EST) algorithm [HKLR02] and the rapidly exploring random tree (RRT) algorithm [LK01]; both of which relied on forward integration of the dynamics with random control inputs instead of a point-to-point local planning subroutine. Further, both methods boasted probabilistic completeness like the PRM algorithm.

The next major development in sampling-based motion planning was optimal variations of the PRM and RRT, denoted PRM* and RRT* [KF11] developed by Karaman and Frazzoli. The innovation was in the selection of edges in the PRM graph. Karaman and Frazzoli described how to construct the graph to be as sparse as possible while ensuring asymptotic optimality with respect to a performance objective in addition to probabilistic completeness. The RRT* algorithm is in fact more closely related to the PRM than the RRT. The RRT* algorithm is essentially an incremental version of the PRM* algorithm which simultaneously constructs a minimum spanning tree in the graph from an initial configuration.

While the RRT* and PRM* algorithms were highly impactful, a local planning

subroutine was once again required. Nonetheless, the utility of the PRM* and RRT* stimulated significant research efforts towards finding general methods for solving the local steering problem. For systems with linear dynamics, classical solutions from linear systems theory can be applied to the local planning subroutine. The drawback to this approach is that the time spent on local planning is prohibitive, taking several minutes for RRT* to produce satisfactory solutions in the examples presented in [WvdB13] and [PPK⁺12]. Additionally, many systems of interest are non-linear making this approach limited in scope.

More recently, the SST algorithm [LLB15] was proposed which provided an algorithm converging asymptotically to an approximately optimal solution without the use of a local planning subroutine. This offers an advantage over RRT* in problems where the local planning subroutine is not available in closed form. However, SST still requires significant running time making it difficult to apply to real-time planning.

1.2 Statement of Contributions

This thesis presents a number of theoretical contributions related to optimal kinodynamic motion planning. The principal contribution is a resolution complete optimal kinodynamic motion planning algorithm based on a direct forward search of the set of input control signals.

Chapters 2 and 3 review the problem formulation addressed, as well as a careful investigation into topological properties of the set of solutions. The key observation in these chapters is continuity of the performance objective with respect to the input signal, presented in Lemma 3.3.3, and a bound on the sensitivity of the cost function with respect to initial conditions, presented in Lemma 3.3.4.

The results of Chapter 3 are the basis of a generalization of classical label correcting algorithms where the comparison of relative cost can be made, not only between trajectories terminating at the same state, but between all trajectories terminating in a particular region of the state space. The generalized label correcting conditions, described in Chapter 5, specify which segments of trajectories can be discarded from the search without compromising convergence to the optimal solution with increasing resolution. The generalized label correcting conditions presented in this thesis are a sharper version of the conditions presented in [PF16] resulting in faster algorithm run-times. Theorem 5.2.1 is a restatement of Bellman's principle of optimality in the context of kinodynamic motion planning and taking into account the topological properties established in Chapter 3. From this result, resolution completeness of the generalized label correcting method follows in Corollary 5.4.1 in the same way that completeness of a label correcting algorithm follows from the principle of optimality in graph search problems.

A wide range of numerical experiments are presented in Chapter 5 which confirm the theoretical results as well as suggest that this method is suitable for real-time planning applications. To further improve the running time of the algorithm, Chapter 6 addresses admissible heuristics for kinodynamic motion planning problems. An admissibility condition for candidate heuristics is presented in Theorem 6.1.1 that

provides a tool for verifying the admissibility of candidate heuristics. Further, this condition characterizes a convex set of admissible heuristics which contains the optimal cost-to-go for a particular problem. An infinite-dimensional linear program is formulated to optimize over the set of admissible heuristics, and it is shown in Theorem 6.2.1 that this linear program is equivalent to the Hamilton-Jacobi-Bellman equation. Lastly, a relaxation of this linear program as a sum-of-squares program is proposed which provides provably admissible heuristics which are as close as possible to the optimal cost-to-go within a finite-dimensional subspace of polynomials.

To create a self-contained thesis, appendices containing a concise review of mathematical analysis, dynamical systems theory, and graph search algorithms are provided at the end of this document.

Chapter References

[AM78] Ralph Abraham and Jerrold E Marsden, *Foundations of mechanics*, 2nd ed., Addison-Wesley, 1978.

[Bet98] John T Betts, *Survey of numerical methods for trajectory optimization*, Journal of guidance, control, and dynamics **21** (1998), no. 2, 193–207.

[BH75] Arthur Bryson and Yu Chi Ho, *Applied optimal control*, Taylor and Francis Publishers, 1975.

[BHTR06] David A Benson, Geoffrey T Huntington, Tom P Thorvaldsen, and Anil V Rao, *Direct trajectory optimization and costate estimation via an orthogonal collocation method*, Journal of guidance, control, and dynamics **29** (2006), no. 6, 1435–1440.

[BKM+96] Jérôme Barraquand, Lydia Kavraki, Rajeev Motwani, Jean-Claude Latombe, Tsai-Yen Li, and Prabhakar Raghavan, *A random sampling scheme for path planning*, Robotics Research, Springer (1996), 249–264.

[BLP85] Rodney A Brooks and Tomas Lozano-Perez, *A subdivision algorithm in configuration space for findpath with rotation*, Transactions on Systems, Man, and Cybernetics, IEEE (1985), no. 2, 224–233.

[Can87] John Canny, *A new algebraic method for robot motion planning and real geometry*, Annual Symposium on the Foundations of Computer Science, IEEE (1987), 39–48.

[HA92] Yong K Hwang and Narendra Ahuja, *A potential field approach to path planning*, Transactions on Robotics and Automation, IEEE **8** (1992), no. 1, 23–32.

[HKLR02] David Hsu, Robert Kindel, Jean-Claude Latombe, and Stephen Rock, *Randomized kinodynamic motion planning with moving obstacles*, The

International Journal of Robotics Research, SAGE Publications **21** (2002), no. 3, 233–255.

[HP87]   Charles R Hargraves and Stephen W Paris, *Direct trajectory optimization using nonlinear programming and collocation*, Journal of guidance, control, and dynamics **10** (1987), no. 4, 338–342.

[KF11]   Sertac Karaman and Emilio Frazzoli, *Sampling-Based Algorithms for Optimal Motion Planning*, The International Journal of Robotics Research, SAGE Publications **30** (2011), no. 7, 846–894.

[Kha86]   Oussama Khatib, *Real-time obstacle avoidance for manipulators and mobile robots*, Autonomous robot vehicles, Springer (1986), 396–404.

[KSLO96]   Lydia E Kavraki, Petr Svestka, Jean-Claude Latombe, and Mark H Overmars, *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*, Transactions on Robotics and Automation, IEEE **12** (1996), no. 4, 566–580.

[LK01]   Steven M LaValle and James J Kuffner, *Randomized kinodynamic planning*, The International Journal of Robotics Research, SAGE Publications **20** (2001), no. 5, 378–400.

[LLB15]   Yanbo Li, Zakary Littlefield, and Kostas E Bekris, *Sparse Methods for Efficient Asymptotically Optimal Kinodynamic Planning*, Algorithmic Foundations of Robotics XI (2015), 263–282.

[PF16]   Brian Paden and Emilio Frazzoli, *A generalized label correcting method for optimal kinodynamic motion planning*, Algorithmic Foundations of Robotics XII (WAFR) (2016), Currently available at: `http://www.wafr.org/program.html#detailedprogram`.

[PPK$^+$12]   Alejandro Perez, Robert Platt, George Konidaris, Leslie Kaelbling, and Tomas Lozano-Perez, *LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics*, International Conference on Robotics and Automation (ICRA), IEEE (2012), 2537–2542.

[PVGBM62]   L. S. Pontryagin, R. V. Gambkrelidze V. G. Boltyanskii, and E. F. Mishchenko, *The mathematical theory of optimal processes*, vol. 4, Gordon and Breach Science Publishers, 1962, English Translation.

[PvY$^+$16]   Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli, *A survey of motion planning and control techniques for self-driving urban vehicles, ieee*, Transactions on Intelligent Vehicles (2016).

[Rei79]   John H Reif, *Complexity of the mover's problem and generalizations*, 421–427.

[RK92]   Elon Rimon and Daniel E Koditschek, *Exact robot navigation using artificial potential functions*, Transactions on robotics and automation, IEEE **8** (1992), no. 5, 501–518.

[SS83]   Jacob T Schwartz and Micha Sharir, *On the "piano movers" problem. II. general techniques for computing topological properties of real algebraic manifolds*, Advances in applied Mathematics **4** (1983), no. 3, 298–351.

[STZ⁺14] Kevin Spieser, Kyle Treleaven, Rick Zhang, Emilio Frazzoli, Daniel Morton, and Marco Pavone, *Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: A case study in Singapore*, Road Vehicle Automation, Springer (2014), 229–245.

[VPO13]   World Health Organization. Violence, Injury Prevention, and World Health Organization, *Global status report on road safety 2013: supporting a decade of action*, World Health Organization, 2013.

[WvdB13] Dustin J Webb and Jur van den Berg, *Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics*, International Conference on Robotics and Automation, IEEE (2013), 5054–5061.

Chapter 2

Problem Formulation

Consider a controlled dynamical system whose state space is $\mathbb{R}^n$, and whose input space is $\mathbb{R}^m$. Canonical state variables for a robot include generalized coordinates and momenta, but can also include relevant quantities such as currents and voltages in electronics. Similarly, canonical input variables for a robot are the generalized forces applied to the robot, but higher fidelity models might include the torque as a state responding to inputs from a drive-train.

To reflect constraints such as obstacle avoidance and velocity limits, the state is restricted to remain in an open[1] subset of *admissible states* X_{free} of $\mathbb{R}^n$ at each instant in time. Similarly, to reflect the design limitations of the actuators, the control is restricted to a bounded subset of *admissible control inputs* Ω of $\mathbb{R}^m$ at each instant in time. Additionally, the motion planning objective is encoded with a terminal constraint that the executed motion terminates in an open set of *goal states* X_{goal} in $\mathbb{R}^n$.

A continuous function x from a closed interval into $\mathbb{R}^n$ is called a *trajectory*. A trajectory is a time history of states over some time interval. If x is a trajectory, then $x(t)$ is the state on that trajectory at time t. A measurable function u from a closed time interval into Ω is called an *input signal*. Like a trajectory, an input signal is a time history, now of control inputs. Instantaneous changes in input are permitted as long as the control signal is mathematically well behaved (measurable). On the other hand, a model of the behavior of a system would not be particularly useful if it permitted instantaneous changes in state, hence the continuity requirement on trajectories (e.g. consider a robot moving instantaneously from one position to another).

---

[1]with respect to the standard topology on $\mathbb{R}^n$.

2.1 Decision Variables

The input signal space $\mathcal{U}$ of all input signals and trajectory space $\mathcal{X}_{x_0}$ of all trajectories starting from the state x_0 are defined

$$\mathcal{U} := \bigcup_{\tau > 0} \{u \in L_1([0,\tau]) : u(t) \in \Omega \ \forall t \in [0,\tau]\}, \qquad (2.1.1)$$

$$\mathcal{X}_{x_0} := \bigcup_{\tau > 0} \{x : [0,\tau] \to \mathbb{R}^n : x(0) = x_0, \quad \mathrm{Lip}(x) \leq M\}, \qquad (2.1.2)$$

where $\mathrm{Lip}(x)$ denotes the greatest lower bound[2] on the set of Lipschitz constants[3] of x. It is important to note that the subscript x_0 in (2.1.2) is a parameter that can be varied to denote the space of trajectories originating from the initial condition x_0. Figure 2-1 illustrates a trajectory with Lipschitz constant M. Since the domains of functions in the sets (2.1.1) and (2.1.2) is variable, it will be useful to denote the *terminal time* of a trajectory's or control signal's domain by $\tau(x)$ or $\tau(u)$ so that the *terminal state* or *terminal control input* is given by $x(\tau(x))$ or $u(\tau(u))$ respectively.

The motivation for defining these two sets comes from the dynamical model of the system[4] given in (2.1.3). The distinguishing feature between various system models is the function f taking a state-control pair into $\mathbb{R}^n$. Throughout this thesis, the function f is assumed to have a known global Lipschitz constant L_f in its first argument, is measurable in its second, and is bounded by a constant M.

$$x(t) = x_0 + \int_{[0,t]} f(x(t), u(t)) \mu(dt). \qquad (2.1.3)$$

A trajectory x is called a *solution* to (2.1.3) for a given input signal $u \in \mathcal{U}$ and initial condition $x_0 \in \mathbb{R}^n$ if it has the same time domain as u and satisfies the equation for each t in the domain of u. The measure μ will refer to the usual Lebesgue

Figure 2-1: The black line illustrates a trajectory x with $\|x\|_{\mathrm{Lip}} \leq M$. The Lipschitz constant bounds the distance between the states at t_1 and t_2 are.

measure[5] on $\mathbb{R}$. Let $\varphi_{x_0} \subset \mathcal{U} \times \mathcal{X}_{x_0}$ be a binary relation between the input signal space and trajectory space defined by (u,x)-pairs where x is a solution to (2.1.3) with input signal u. Theorem 2.1.1 below states that the relation φ_{x_0} is a function mapping $\mathcal{U}$ into $\mathcal{X}_{x_0}$. This will be important to the analysis presented in later chapters.

---

[2] Appendix A.3.2 defines *greatest lower bound*.

[3] Appendix B.2.1 defines *Lipschitz continuity*.

[4] Taking the derivative of (2.1.3) results in the more familiar diffrential form which is equivalent to the integral form.

[5] A review of Lebesgue integration is presented in Appendix C.

Theorem 2.1.1. *For each input signal in* $\mathcal{U}$*, there is a unique solution to* (2.1.3) *in* $\mathcal{X}_{x_0}$.

The function φ_{x_0} will be called the *system map* in light of this observation. This is a straight forward corollary to the standard existence-uniqueness theorem for integral equations provided in Appendix C.2.

2.2 Problem Specifications

We will denote the subset of trajectories which originate from a given *initial condition* x_{ic} and remain within the set X_{free} by $\mathcal{X}_{\text{free}}$.

$$\mathcal{X}_{\text{free}} := \left\{ x \in \mathcal{X}_{x_{ic}} : \ x(t) \in X_{\text{free}} \ \forall t \in [0, \tau(x)] \right\}. \qquad (2.2.1)$$

The subset of trajectories in $\mathcal{X}_{\text{free}}$ which additionally terminate in X_{goal} is denoted $\mathcal{X}_{\text{goal}}$.

$$\mathcal{X}_{\text{goal}} := \left\{ x \in \mathcal{X}_{\text{free}} : \ x(\tau(x)) \in X_{\text{goal}} \right\}. \qquad (2.2.2)$$

Figure 2-2 illustrates example trajectories in $\mathcal{X}_{\text{goal}}$ and $\mathcal{X}_{\text{free}}$.

The subset of input signals in $\mathcal{U}$ such that $\varphi_{x_{ic}}(u) \in \mathcal{X}_{\text{free}}$ is denoted $\mathcal{U}_{\text{free}}$. Similarly, the subset of input signals $\mathcal{U}$ such that $\varphi_{x_{ic}}(u) \in \mathcal{X}_{\text{goal}}$ is denoted $\mathcal{U}_{\text{goal}}$. It is important to remark that $\mathcal{U}_{\text{free}}$ and $\mathcal{U}_{\text{goal}}$ are the inverse images of $\mathcal{X}_{\text{free}}$ and $\mathcal{X}_{\text{goal}}$ under the system map $\varphi_{x_{ic}}$,

$$\mathcal{U}_{\text{free}} = \varphi_{x_{ic}}^{-1}(\mathcal{X}_{\text{free}}), \qquad \mathcal{U}_{\text{goal}} = \varphi_{x_{ic}}^{-1}(\mathcal{X}_{\text{goal}}). \qquad (2.2.3)$$

The first part of the problem specification is to find a signal $u \in \mathcal{U}_{\text{goal}}$.

Since there are often many input signals in $\mathcal{U}_{\text{goal}}$, we additionally would like to minimize a performance objective. Performance objectives of the form

$$J_{x_0}(u) = \int_{[0, \tau(u)]} g([\varphi_{x_0}(u)](t), u(t)) \, \mu(dt), \qquad (2.2.4)$$

Figure 2-2: The admissible states X_{free} consist of the rectangular region excluding the gray obstacles. The goal region X_{goal} is the green circle. $\tilde{x}$ depicts a trajectory in $\mathcal{X}_{\text{goal}}$ while x shows a trajectory in $\mathcal{X}_{\text{free}}$.

are considered. The function g from $X_{\text{free}} \times \Omega$ into $(0, \infty)$ is a *running cost* which gives a rate of accumulation of cost for each state-control pair. It is assumed that a global Lipschitz constant L_g for g on $X_{\text{free}} \times \Omega$ is known.

Since the state trajectory is completely determined by the input signal and initial condition, the cost functional J is considered a mapping from $\mathcal{U}$ into $(0, \infty)$ parameterized by the initial condition. It will be convenient to extend the domain of J_{x_0} with
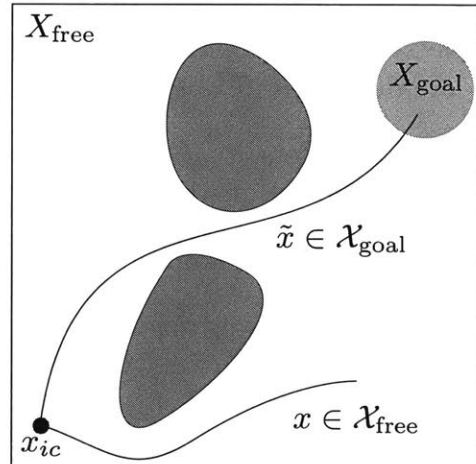
a symbol NULL where $J_{x_0}(\text{NULL}) = \infty$ for every x_0. We will also adopt the convention that $\inf_{u \in \emptyset} J_{x_{ic}}(u) = \infty$.

Intuitively, we would like to minimize the cost J over the set $\mathcal{U}_{\text{goal}}$. However, a particular problem instance may not admit a minimizer or even a solution at all. Therefore, we will seek a solution to the following *relaxed optimal kinodynamic motion planning problem*:

Problem 2.2.1. *Given an initial condition x_{ic}, a set of admissible states X_{free}, a set of goal states X_{goal}, a set of admissible control inputs Ω, a dynamic model f, and a running cost g; find a sequence $\{u_R\} \subset \mathcal{U}_{\text{goal}} \cup \text{NULL}$ such that*

$$\lim_{R \to \infty} J_{x_{ic}}(u_R) = \inf_{u \in \mathcal{U}_{\text{goal}}} J_{x_{ic}}(u) := c^*. \tag{2.2.5}$$

An algorithm parameterized by a resolution $R \in \mathbb{N}$ whose output for each R forms a sequence solving this problem will be called *resolution complete*.

2.3 Review of Assumptions

To summarize the problem formulation, the *problem data* for the optimal kinodynamic motion planning problem is $(x_{ic}, X_{\text{free}}, X_{\text{goal}}, \Omega, f, g)$, where x_{ic} is an initial condition, X_{free} is the set of admissible states, X_{goal} is the set of goal states, f is the model of the system dynamics, and g is the running cost. The assumptions on the problem data described in this section are:

A-1 The sets X_{free} and X_{goal} are open with respect to the topology induced by the Euclidean distance.

A-2 The set of admissible inputs Ω is bounded, and a bound $u_{\max}$ is known.

A-3 The running cost g is strictly positive and Lipschitz continuous in both arguments with a known Lipschitz constant L_g.

A-4 The dynamic model f is Lipschitz continuous in its first argument with a known Lipschitz constant L_f, Lebesgue measurable in its second argument, and bounded in both arguments by M.

There are two important things to consider regarding the assumptions of the input data to an algorithm or method. First, the applicability of a method requiring the assumptions to be met depends on how general the assumptions are. Secondly, given an instance of problem data, discerning whether it meets the assumptions should be an easy decision problem, decidable with an algorithm of lesser complexity than the algorithm the problem data is being given to. Otherwise, there is little value to the algorithm.

The formulation and assumptions of this chapter are quite general and can be discerned by inspection from typical problem data. For example, if X_{free} is described by the union of a finite set of closed polyhedra and X_{goal} is the union of a set of

open polyhedra, then A-1 is satisfied. Similarly, robot actuators have clear design limitations bounding the set of admissible inputs so that A-2 is satisfied in all practical instances. Assumption A-3 is often verified by inspection of the running cost g. For example, minimum time problems with $g(z, w) = 1$ are among the most frequently discussed objectives. The Lipschitz constant in this case is 0. Lipschitz continuity of the dynamics is generally required for dynamical models to be well defined. Therefore, most models derived from physical laws will satisfy A-4. As a final remark on the generality of this formulation, it does not require any controllability properties of the dynamic model f as is the case in [LLB15, KF10]. Although this makes the formulation more general it does not necessarily add to the practical value since there are few applications where motion planning is used on an uncontrollable system.

2.3.1 Verifiability of Assumptions

Most well known sampling-based motion planning algorithms impose an abstract assumption on the problem data for the algorithm to have the desired theoretical guarantees. In [KKL98] the *ε-goodness* property was the basis of an analysis of the PRM algorithm. In [HLM97] the *expansiveness* property was required of the problem data to prove the probabilistic completeness of the EST algorithm. Similarly, in [LK01] the existence of an *attraction sequence* is required to prove the probabilistic completeness of the RRT algorithm. Lastly, in [KF11], probabilistic completeness of the RRT* algorithm relies on the problem data satisfying a *δ-robustness* property.

While these assumptions are precisely defined, it is not clear how to verify if these properties are satisfied, whether this subset of problem instances are of practical value, or if there even exists a problem instance satisfying a particular assumption. These are important open problems in motion planning that have received little attention.

The following chapter uses techniques from the subject of topology to develop a foundation of tools for analyzing the optimal kinodynamic motion planning problem. A brief introduction to topology is presented in Appendix B.1.

To motivate the use of these techniques, we apply them to formally prove the intuitive fact that an open set of admissible states is sufficient for the δ-robustness assumption of the RRT* algorithm. The RRT* algorithm requires that feasible problem instances be *δ-robustly feasible*. That is, feasible problem instances must have a trajectory $x \in \mathcal{X}_{\text{goal}}$, such that for some $\delta > 0$ the trajectory satisfies

$$B_\delta(x(t)) \subset X_{\text{free}}, \quad \forall t \in [0, \tau(x_1)]. \tag{2.3.1}$$

Lemma 2.3.1. *If X_{free} is open, then every feasible problem instance ($\mathcal{X}_{\text{goal}} \neq \emptyset$) is δ-robustly feasible.*

Proof. Let x be a trajectory in $\mathcal{X}_{\text{goal}}$. Since x is continuous, $x([0, \tau(x)])$ is a compact subset of X_{free} for all t in the interval $[0, \tau(x)]$ (cf. Lemma B.1.5). By assumption, X_{free} is open so there exists a $\delta > 0$ such that $B_\delta(x(t)) \subset X_{\text{free}}$ for all t in $[0, \tau(x)]$ (cf. Corollary B.2.1) which is the definition of δ-robust feasibility.

$\square$

Chapter References

[HLM97] David Hsu, J-C Latombe, and Rajeev Motwani, *Path planning in expansive configuration spaces*, International Conference on Robotics and Automation, vol. 3, IEEE, 1997, pp. 2719–2726.

[KF10] Sertac Karaman and Emilio Frazzoli, *Optimal kinodynamic motion planning using incremental sampling-based methods*, Conference on Decision and Control, IEEE, 2010, pp. 7681–7687.

[KF11] Sertac Karaman and Emilio Frazzoli, *Sampling-based algorithms for optimal motion planning*, The international journal of robotics research, Sage Publications **30** (2011), no. 7, 846–894.

[KKL98] Lydia E Kavraki, Mihail N Kolountzakis, and J-C Latombe, *Analysis of probabilistic roadmaps for path planning*, Transactions on Robotics and Automation, IEEE **14** (1998), no. 1, 166–171.

[LK01] Steven M LaValle and James J Kuffner, *Randomized kinodynamic planning*, The International Journal of Robotics Research, SAGE Publications **20** (2001), no. 5, 378–400.

[LLB15] Yanbo Li, Zakary Littlefield, and Kostas E Bekris, *Sparse methods for efficient asymptotically optimal kinodynamic planning*, Algorithmic Foundations of Robotics XI, Springer, 2015, pp. 263–282.

Chapter 3

Topological Properties of the Input Signal and Trajectory Spaces

The problem formulation will use concepts from general topology to establish basic properties used by the GLC method.

The function spaces $\mathcal{X}_{x_{ic}}$ and $\mathcal{U}$ are equipped with metrics in order to perform an analysis on the subsets $\mathcal{X}_{\text{free}}$, $\mathcal{X}_{\text{goal}}$, $\mathcal{U}_{\text{free}}$, and $\mathcal{U}_{\text{goal}}$ in their respective metric topologies. It will follow from the assumption that X_{free} and X_{goal} are open in the standard topology on $\mathbb{R}^n$ that $\mathcal{X}_{\text{free}}$ and $\mathcal{X}_{\text{goal}}$ are open in the metric topology on $\mathcal{X}_{x_{ic}}$. We then review a result based on Gronwall's inequality[1] which shows $\varphi_{x_{ic}}$ is a continuous mapping from $\mathcal{U}$ into $\mathcal{X}_{x_{ic}}$. A direct consequence of these observations is $\mathcal{U}_{\text{free}}$ and $\mathcal{U}_{\text{goal}}$ are open subsets of $\mathcal{U}$ since the inverse image of an open set under a continuous function is open[2]. The final important observation is the continuity of the cost function from $\mathcal{U}$ into $\mathbb{R}$.

These results will be used in later chapters as follows: A dense subset of $\mathcal{U}$ will be constructed in Chapter 4 as an approximation of $\mathcal{U}$. Since $\mathcal{U}_{\text{free}}$ and $\mathcal{U}_{\text{goal}}$ are open in $\mathcal{U}$, the approximation will also be dense in $\mathcal{U}_{\text{free}}$ and $\mathcal{U}_{\text{goal}}$. Next, the image of the approximation, intersected with $\mathcal{U}_{\text{free}}$ and $\mathcal{U}_{\text{goal}}$, under the cost function $J_{x_{ic}}$ will be dense in $J_{x_{ic}}(\mathcal{U}_{\text{free}})$ and $J_{x_{ic}}(\mathcal{U}_{\text{goal}})$ so that a signal, as will be seen, in the approximation of $\mathcal{U}_{\text{goal}}$ achieves the optimal cost with arbitrarily high accuracy. Lastly, in Chapter 5 the continuity of the cost function is used once again to identify a large subset of the approximation which can be discarded without compromising the accuracy with which it approximates a signal with the optimal cost.

---

[1]Gronwall's inequality is discussed in Appendix C.

[2]This is the notion of continuity for functions on topological spaces. The relation to the definition for metric spaces is discussed in Lemma B.2.2.
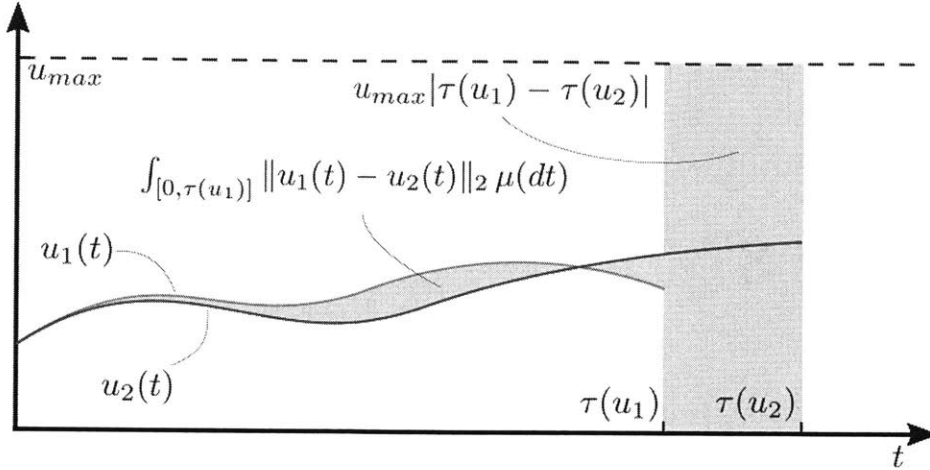
Figure 3-1: The distance $d_{\mathcal{U}}(u_1, u_2)$ between two signals is illustrated by the region shaded in red. Note that the signals have different time domains, and that the second term in (3.1.1) describes the worst-case difference on the interval $[\tau(u_1), \tau(u_2)]$ between the two signals if u_1 were extended to have the same domain as u_2.

3.1   Metrics on the Signal and Trajectory Spaces

The input signal and trajectory spaces become metric spaces when equipped with the metrics $d_{\mathcal{U}}$ and $d_{\mathcal{X}_{x_0}}$ adapted from [YL11]:

$$d_{\mathcal{U}}(u_1, u_2) := \int_{[0, \min\{\tau(u_1), \tau(u_2)\}]} \|u_1(t) - u_2(t)\|_2 \, \mu(dt) + u_{max}|\tau(u_1) - \tau(u_2)|. \quad (3.1.1)$$

$$d_{\mathcal{X}}(x_1, x_2) := \max_{t \in [0, \min\{\tau(x_1), \tau(x_2)\}]} \{\|x_1(t) - x_2(t)\|\} + M|\tau(x_1) - \tau(x_2)|. \quad (3.1.2)$$

Figures 3-1 and 3-2 illustrate these two metrics. Several results presented in this chapter regarding these metrics are simple variations of results in [YL11]. However, appropriately modified proofs are provided for the version of these results needed in later chapters.

While these distance functions have been utilized as metrics in the literature, verification that they satisfy the definition of a metric have not been published. These proofs are provided in Appendix E.

28
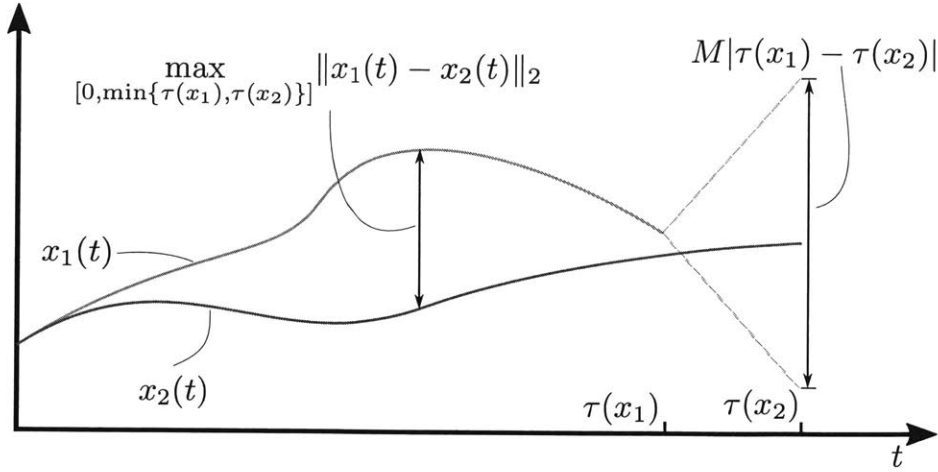
Figure 3-2: The two terms making up the expression for $d_{\mathcal{X}}(x_1, x_2)$ are illustrated by two annotated distances. Observe that the second term in (3.1.2) describes the worst case distance between the two if x_1 were extended to have the same domain as x_2.

3.2 Continuity of the System Map φ_{x_0} and Cost Functional J_{x_0}
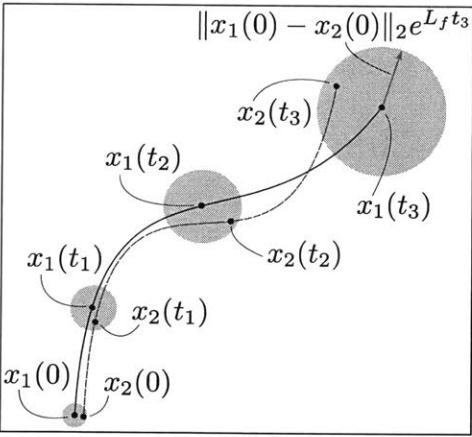
Figure 3-3: Illustration of Lemma 3.2.1. The distance of $x_2(t)$ from $x_1(t)$ is bounded by the initial distance multiplied by $e^{L_f t}$. The circles illustrate possible locations of each $x_2(t_i)$ relative to $x_1(t_i)$ predicted by the inequality.

A classical result in dynamical systems theory is the continuity of trajectories with respect to initial conditions. Lemma 3.2.1 is a restatement of this result with the appropriate notation.

Lemma 3.2.1. *Let u be an input signals in $\mathcal{U}$; x_0, z_0 initial conditions in $\mathbb{R}^n$; and $x_1 := \varphi_{x_0}(u)$, $x_2 := \varphi_{z_0}(u)$ the corresponding trajectories. Then the difference between these trajectories satisfies the inequality*

$$\|x_1(t) - x_2(t)\|_2 \leq \|x_0 - z_0\|_2 e^{L_f t}. \qquad (3.2.1)$$

The standard proof can be found in Appendix C.2. Figure 3-3 illustrates the inequality in (3.2.1).

Lemma 3.2.2 ([YL11]). *The system map φ_{x_0} is continuous from $(\mathcal{U}, d_{\mathcal{U}})$ into $(\mathcal{X}_{x_0}, d_{\mathcal{X}})$.*

Proof. Let u_1, u_2 be input signals in $\mathcal{U}$, ordered so that $\tau(u_1) \leq \tau(u_2)$. We will show that the map φ_{x_0} is continuous at u_1. Denote trajectories $\varphi_{x_0}(u_1)$ and $\varphi_{x_0}(u_2)$ by x_1 and x_2

respectively. For t in the time interval $[0, \tau(u_1)]$, it follows from equation (2.1.3) that

$$\|x_1(t) - x_2(t)\|_2 = \left\| \int_{[0,t]} f(x_1(\tau), u_1(\tau))\, \mu(d\tau) - \int_{[0,t]} f(x_2(\tau), u_2(\tau))\, \mu(d\tau) \right\|_2$$

$$\leq \int_{[0,t]} \|f(x_1(\tau), u_1(\tau)) - f(x_2(\tau), u_2(\tau))\|_2\, \mu(d\tau).$$

(3.2.2)

Then using the Lipschitz continuity of the dynamic model in A-4,

$$\|x_1(t) - x_2(t)\|_2 \leq \int_{[0,t]} L_f \|x_1(\tau) - x_2(\tau)\|_2 + L_f \|u_1(\tau) - u_2(\tau)\|_2\, \mu(d\tau)$$

$$\|x_1(t) - x_2(t)\|_2 \leq \int_{[0,t]} L_f \|x_1(\tau) - x_2(\tau)\|_2\, \mu(d\tau) + L_f \int_{[0,t]} \|u_1(\tau) - u_2(\tau)\|_2\, \mu(d\tau)$$

$$\leq L_f d_{\mathcal{U}}(u_1, u_2) + \int_{[0,t]} L_f \|x_1(\tau) - x_2(\tau)\|_2\, \mu(d\tau).$$

(3.2.3)

Then by Gronwall's inequality (cf. Lemma C.2.1),

$$\|x_1(t) - x_2(t)\|_2 \leq L_f d_{\mathcal{U}}(u_1, u_2) e^{L_f t}.$$

(3.2.4)

On the remaining time domain $[\tau(u_1), \tau(u_2)]$, observe that $d_{\mathcal{U}}(u_1, u_2) < \delta$ implies $|\tau(u_1) - \tau(u_2)| < d_{\mathcal{U}}(u_1, u_2)/u_{max}$. Therefore,

$$\begin{aligned} M|\tau(x_1) - \tau(x_2)| &= M|\tau(u_1) - \tau(u_2)| \\ &< M d_{\mathcal{U}}(u_1, u_2)/u_{max}. \end{aligned}$$

(3.2.5)

Then $d_{\mathcal{X}}(x_1, x_2)$ can be bounded as a function of δ by combining (3.2.4) and (3.2.5),

$$\begin{aligned} d_{\mathcal{X}}(x_1, x_2) &= \max_{t \in [0, \min\{\tau(x_1), \tau(x_2)\}]} \{\|x_1(t) - x_2(t)\|_2\} + M|\tau(x_1) - \tau(x_2)| \\ &< L_f d_{\mathcal{U}}(u_1, u_2) e^{L_f \tau(u_1)} + M d_{\mathcal{U}}(u_1, u_2)/u_{max} \\ &= (L_f e^{L_f \tau(u_1)} + M/u_{max}) d_{\mathcal{U}}(u_1, u_2). \end{aligned}$$

(3.2.6)

Thus, the map φ_{x_0} continuous at u_1 since for any $\varepsilon > 0$, the signal u_2 can be chosen sufficiently close to u_1 so that $d_{\mathcal{X}}(x_1, x_2) < \varepsilon$, which is equivalent to the definition of continuity (cf. Lemma B.2.2). $\qquad\square$

Note that the continuity is not uniform because of the appearance of $\tau(u_1)$ in (3.2.6).

3.3 Properties of the Set of Solutions

The first important observation is that $\mathcal{X}_{\text{free}}$ and $\mathcal{X}_{\text{goal}}$ are open in the metric topology defined by $d_{\mathcal{X}}$ when X_{free} and X_{goal} are open in the standard topology.

Lemma 3.3.1 ([YL11]). *$\mathcal{X}_{\text{free}}$ is an open subset of $\mathcal{X}_{x_{ic}}$ in the topology induced by $d_{\mathcal{X}}$.*
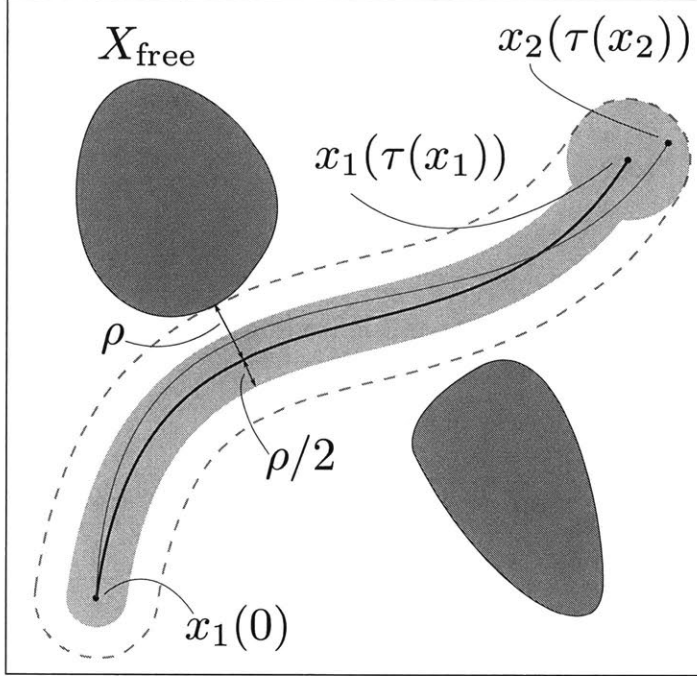
Figure 3-4: Illustration of the trajectories x_1 and x_2 discussed in Lemma 3.3.1. Any trajectory x_2 that is within a distance (in the sense of $d_{\mathcal{X}}$) of $\rho/2$ of x_1, by definition, remains in the red region around x_1.

Proof. Either $\mathcal{X}_{\text{free}}$ is empty, in which case it is open (cf. Appendix B.1), or it is non-empty. Assume the latter case and let x_1 be an element of $\mathcal{X}_{\text{free}}$. Since x_1 is a Lipschitz continuous function from the compact interval $[0, \tau(x)]$ into $\mathbb{R}^n$, its image, $x_1([0, \tau(x)])$, is also compact (cf. Lemma B.1.5). Next, since X_{free} is open, and $x([0, \tau(x_1)])$ is compact, there exists a $\rho > 0$ such that $B_\rho(x_1(t)) \subset X_{\text{free}}$ for all $t \in [0, \tau(x_1)]$ (cf. Corollary B.2.1). Now consider a trajectory x_2 in the ball of radius $\rho/2$ centered at x_1 in $\mathcal{X}_{x_{ic}}$ (note that this is in the function space $\mathcal{X}_{x_{ic}}$). Then, from the definition of $d_{\mathcal{X}}$,

$$\max_{t \in [0, \min\{\tau(x_1), \tau(x_2)\}]} \{\|x_1(t) - x_2(t)\|_2\} + M|\tau(x_1) - \tau(x_2)| < \rho/2. \tag{3.3.1}$$

The first term in (3.3.1) implies, that on the interval $[0, \min\{\tau(x_1), \tau(x_2)\}]$, the distance between $x_1(t)$ and $x_2(t)$ is within $\rho/2$. If $\tau(x_2) \le \tau(x_1)$, then x_2 is necessarily in $\mathcal{X}_{\text{free}}$. Now suppose that $\tau(x_1) < \tau(x_2)$. Then for $t > \tau(x_1)$,

$$\|x_2(t) - x_1(\tau(x_1))\|_2 \le \|x_1(\tau(x_1)) - x_2(\tau(x_1))\|_2 + \|x_2(\tau(x_1)) - x_2(t)\|_2. \tag{3.3.2}$$

The last term in (3.3.2) is further bounded by the Lipschitz constant for x_2,

$$\|x_2(t) - x_1(\tau(x_1))\|_2 \le \|x_1(\tau(x_1)) - x_2(\tau(x_1))\|_2 + M|t - \tau(x_1)|$$
$$\le \|x_1(\tau(x_1)) - x_2(\tau(x_1))\|_2 + M|\tau(x_2) - \tau(x_1)| \tag{3.3.3}$$
$$\le \rho.$$

Thus, at any time t, $x_2(t)$ is within a distance of ρ from some point in $x_1([0, \tau(x_1)])$. Therefore, x_2 is in $\mathcal{X}_{\text{free}}$. Since the choice of x_2 was arbitrary, x_1 is an interior point of $\mathcal{X}_{\text{free}}$; since the choice of x_1 was arbitrary, $\mathcal{X}_{\text{free}}$ is open.

$\square$

The trajectories discussed in the proof are illustrated in Figure 3-4. An analogous observation holds for the set $\mathcal{X}_{\text{goal}}$.

Lemma 3.3.2 ([YL11]). $\mathcal{X}_{\text{goal}}$ *is an open subset of* $\mathcal{X}_{x_{ic}}$ *with respect to the topology induced by* $d_{\mathcal{X}}$.

Proof. Either $\mathcal{X}_{\text{goal}}$ is empty, in which case it is open, or it is non-empty. Assume the latter case and let x_1 be a trajectory in $\mathcal{X}_{\text{goal}}$ in the metric subspace $(\mathcal{X}_{\text{free}}, d_{\mathcal{X}})$. Since X_{goal} is open, the terminal point $x_1(\tau(x_1))$ is an interior point of X_{goal}, so there exists a $\rho > 0$ such that $B_{\rho/2}(x_1(\tau(x_1)))$ is a subset of X_{goal}. Now consider a trajectory x_2 such that $d_{\mathcal{X}}(x_1, x_2) < \rho/2$. From the definition of $d_{\mathcal{X}}$,

$$\max_{t \in [0, \min\{\tau(x_1), \tau(x_2)\}]} \{\|x_1(t) - x_2(t)\|_2\} + M|\tau(x_1) - \tau(x_2)| < \rho/2. \qquad (3.3.4)$$

From (3.3.4), the difference in terminal times of x_1 and x_2 satisfies the bound

$$|\tau(x_1) - \tau(x_2)| < \rho/(2M). \qquad (3.3.5)$$

Similarly, the distance between the states of the two trajectories at the shorter of the terminal times is bounded by

$$\|x_1(\min\{\tau(x_1), \tau(x_2)\}) - x_1(\min\{\tau(x_1), \tau(x_2)\})\|_2 < \rho/2. \qquad (3.3.6)$$

These observations are then used to bound the distance between the terminal states of the two trajectories. The next step requires considering two cases: First, suppose $\tau(x_2) \leq \tau(x_1)$. Then

$$\|x_2(\tau(x_2)) - x_1(\tau(x_1))\|_2 \leq \|x_2(\tau(x_2)) - x_1(\tau(x_2))\|_2 + \|x_1(\tau(x_1)) - x_1(\tau(x_2))\|_2$$
$$\leq \rho/2 + M|\tau(x_2) - \tau(x_1)|$$
$$\leq \rho/2 + \frac{M\rho}{2M}$$
$$\leq \rho.$$

$$(3.3.7)$$

Second, suppose $\tau(x_1) < \tau(x_2)$. The analogous application of the triangle inequality

yields,

$$\|x_2(\tau(x_2)) - x_1(\tau(x_1))\|_2 \leq \|x_2(\tau(x_1)) - x_1(\tau(x_1))\|_2 + \|x_2(\tau(x_1)) - x_2(\tau(x_2))\|_2$$

$$\leq \rho/2 + M|\tau(x_2) - \tau(x_1)|$$

$$\leq \rho/2 + \tfrac{M\rho}{2M}$$

$$\leq \rho.$$

(3.3.8)

Thus, the terminal state $x_2(\tau(x_2))$ is an element of X_{goal}. Since the choice of x_2 was arbitrary, x_1 is an interior point of $\mathcal{X}_{\text{goal}}$; since the choice of x_1 was arbitrary, every point in $\mathcal{X}_{\text{goal}}$ is an interior point. Thus, $\mathcal{X}_{\text{goal}}$ is open in the metric subspace $(\mathcal{X}_{\text{free}}, d_{\mathcal{X}})$. Since $\mathcal{X}_{\text{free}}$ is open $\mathcal{X}_{x_{ic}}$, $\mathcal{X}_{\text{goal}}$ is open in $\mathcal{X}_{x_{ic}}$ (cf. Lemma B.1.1). $\qquad\square$

Now that we have established that the system map is continuous and $\mathcal{X}_{\text{free}}$ and $\mathcal{X}_{\text{goal}}$ are open subsets of its codomain, by definition of continuity, the inverse images of $\mathcal{X}_{\text{free}}$ and $\mathcal{X}_{\text{goal}}$ under the system map are open.

Theorem 3.3.1 ([YL11]). $\mathcal{U}_{\text{free}}$ and $\mathcal{U}_{\text{goal}}$ are open in the topology induced by $d_{\mathcal{U}}$.

Proof. By Lemma 3.2.1 the map $\varphi_{x_{ic}}$ is continuous, and by Lemmas 3.3.1 and 3.3.2, $\mathcal{X}_{\text{free}}$ and $\mathcal{X}_{\text{goal}}$ are open. The sets $\mathcal{U}_{\text{free}}$ and $\mathcal{U}_{\text{goal}}$ are defined as

$$\mathcal{U}_{\text{free}} = \varphi_{x_{ic}}^{-1}(\mathcal{X}_{\text{free}}), \qquad \mathcal{U}_{\text{goal}} = \varphi_{x_{ic}}^{-1}(\mathcal{X}_{\text{goal}}).$$

(3.3.9)

By the definition of continuity[3], $\mathcal{U}_{\text{free}}$ and $\mathcal{U}_{\text{goal}}$ are open. $\qquad\square$

This is the essential property needed to prove the resolution completeness of the algorithm presented in later chapters.

Next, similar continuity properties for the cost function are derived. This continuity will be used to argue that the image of a dense subset of $\mathcal{U}$ will be dense in the cost space.

Lemma 3.3.3. $J_{x_0} : \mathcal{U} \to \mathbb{R}$ is continuous for any $x_0 \in \mathbb{R}^n$.

Proof. Let $u_1, u_2 \in \mathcal{U}$ and, without loss of generality, assume $\tau(u_1) \leq \tau(u_2)$. Denote trajectories $\varphi_{x_0}(u_1)$ and $\varphi_{x_0}(u_2)$ by x_1 and x_2 respectively. The associated difference in cost is

$$|J_{x_0}(u_1) - J_{x_0}(u_2)| = \left| \int_{[0,\tau(u_1)]} g(x_1(t), u_1(t))\, \mu(dt) - \int_{[0,\tau(u_2)]} g(x_2(t), u_2(t))\, \mu(dt) \right|$$

$$\leq \left| \int_{[0,\tau(u_1)]} g(x_1(t), u_1(t)) - g(x_2(t), u_2(t))\, d\mu(t) \right.$$

$$\left. - \int_{[\tau(u_1),\tau(u_2)]} g(x_2(t), u_2(t))\, d\mu(t) \right|.$$

$$\leq \int_{[0,\tau(u_1)]} |g(x_1(t), u_1(t)) - g(x_2(t), u_2(t))|\, \mu(dt)$$

$$+ \int_{[\tau(u_1),\tau(u_2)]} |g(x_2(t), u_2(t))|\, \mu(dt).$$

(3.3.10)

---

[3]Continuity is defined in Appendix B.1

33

This is further bounded using the the Lipschitz constant of g (cf. A-3)

$$
\begin{aligned}
|J_{x_0}(u_1) - J_{x_0}(u_2)| &\leq \int_{[0,\tau(u_1)]} L_g \|x_1(t) - x_2(t)\|_2 + L_g \|u_1(t) - u_2(t)\|_2 \ \mu(dt) \\
&\quad + \int_{[\tau(u_1),\tau(u_2)]} g\left(x_2(t), u_2(t)\right) \mu(dt) \\
&\leq L_g \tau(u_1) d_\mathcal{X}(x_1, x_2) + L_g d_\mathcal{U}(u_1, u_2) \\
&\quad + \int_{[\tau(u_1),\tau(u_2)]} g\left(x_2(t), u_2(t)\right) \mu(dt).
\end{aligned}
$$
(3.3.11)

Since x_2 is continuous, u_2 is bounded, and g is continuous, there exists a bound G on $g(x_2(t), u_2(t))$ for $t \in [\tau(u_1), \tau(u_2)]$. Thus, the difference in cost is further bounded by

$$
|J_{x_0}(u_1) - J_{x_0}(u_2)| \leq L_g \tau(u_1) d_\mathcal{X}(x_1, x_2) + L_g d_\mathcal{U}(u_1, u_2) + G|\tau(u_2) - \tau(u_1)|.
$$
(3.3.12)

Observe that $|\tau(u_2) - \tau(u_1)| < d_\mathcal{U}(u_1, u_2)/u_{max}$ so the term $G|\tau(u_1) - \tau(u_2)|$ can be made arbitrarily small with $d_\mathcal{U}(u_1, u_2)$ sufficiently small. Similarly, since φ_{x_0} is continuous, $d_\mathcal{X}(x_1, x_2)$ is made arbitrarily small with $d_\mathcal{U}(u_1, u_2)$ sufficiently small. Thus, for any $\varepsilon > 0$, there exists a $\delta > 0$ such that $d_\mathcal{U}(u_1, u_2) < \delta$ implies $|J_{x_0}(u_1) - J_{x_0}(u_2) < \varepsilon|$ which implies continuity by Lemma B.2.2. $\square$

The cost functional is continuous with respect to the initial condition parameter as well. In fact, it is Lipschitz continuous which is used when comparing the cost of two trajectories to check if one of the trajectories is provably suboptimal.

Lemma 3.3.4. *For any $u \in \mathcal{U}$ and $x_0, z_0 \in \mathbb{R}^n$,*

$$
|J_{x_0}(u) - J_{z_0}(u)| \leq \|x_0 - z_0\|_2 \cdot \frac{L_g}{L_f}\left(e^{L_f \tau(u)} - 1\right)
$$
(3.3.13)

Proof. The absolute difference is bounded using the Lipschitz continuity of g. This is further bounded using (3.2.1). Denoting $x(t) = [\varphi_{x_0}(u)](t)$ and $z(t) = [\varphi_{z_0}(u)](t)$,

$$
\begin{aligned}
|J_{x_0}(u) - J_{z_0}(u)| &= \left| \int_{[0,\tau(u)]} g(x(t), u(t)) - g(z(t), u(t)) \, d\mu(t) \right| \\
&\leq \int_{[0,\tau(u)]} |g(x(t), u(t)) - g(z(t), u(t))| \ \mu(dt) \\
&\leq \int_{[0,\tau(u)]} L_g \|x(t) - z(t)\|_2 \ \mu(dt) \\
&\leq \int_{[0,\tau(u)]} \|x_0 - z_0\|_2 L_g e^{L_f t} \ \mu(dt) \\
&= \|x_0 - z_0\|_2 \frac{L_g}{L_f}\left(e^{L_f \tau(u)} - 1\right).
\end{aligned}
$$
(3.3.14)

$\square$

Chapter References

[YL11] Dmitry S Yershov and Steven M LaValle, *Sufficient conditions for the existence of resolution complete planning algorithms*, Algorithmic Foundations of Robotics IX, Springer (2011), 303–320.

Chapter 4

Approximation of the Space of Input Signals

The goal of this thesis is to present an algorithm which computes an approximately optimal solution to Problem 2.2.1, and, since the decision variable is selected from $\mathcal{U}$, an uncountably infinite space, we will use a countably infinite approximation which can be systematically searched to provide an approximate solution with arbitrary accuracy in finite time.

The natural number R will denote the resolution of the approximation, and the signal space $\mathcal{U}$ is approximated by a subset $\mathcal{U}_R$ indexed by the resolution. The approximation $\mathcal{U}_R$ of $\mathcal{U}$ is constructed from strings of a finite collection of primitive input signals. The primitive input signals each take a constant value from a finite subset Ω_R of Ω for a fixed duration.

Figure 4-1: The set Ω is illustrated in light red with the subset Ω_R indicated by small black markers. The dispersion of Ω_R in Ω is illustrated by the radius of the dark red circle. It is the radius of the largest ball centered at a point in Ω which does not intersect Ω_R.

The approximation of the control inputs Ω can be any sequence of subsets whose *dispersion* converges to zero in Ω. That is,

$$\lim_{R \to \infty} \left(\sup_{w \in \Omega} \inf_{v \in \Omega_R} \|w - v\|_2 \right) = 0. \tag{4.0.1}$$

The dispersion of a subset within a set is a measure of how well the subset approximates its containing set. Figure 4-1 illustrates the definition of dispersion appearing in equation (4.0.1). A family of subsets Ω_R exists and is often easily obtained with regular grids or random sampling for a given Ω.

The primitive input signal $u : [0, 1/R] \to \Omega$ associated to each w in Ω_R is the
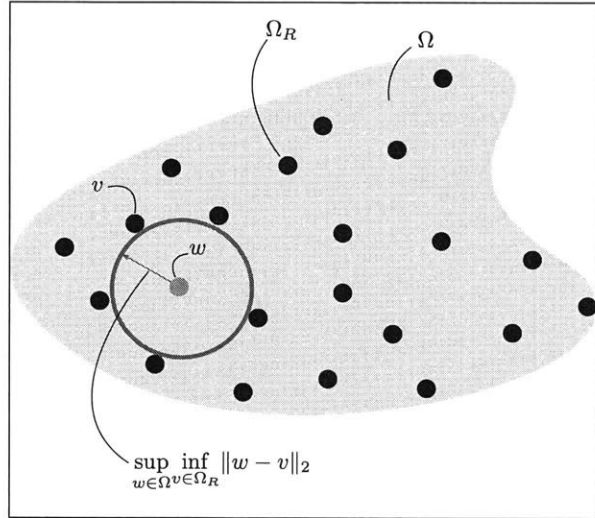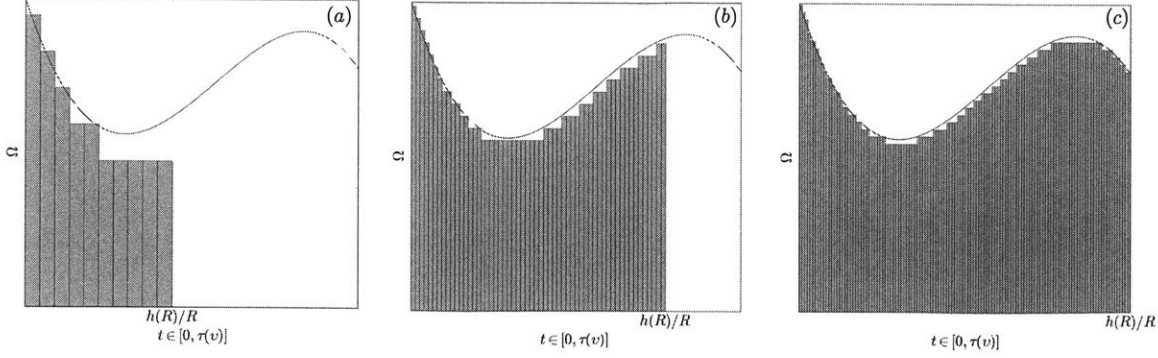
Figure 4-3: The three tiles illustrate how, with increasing resolution (left to right), there is a signal in $\mathcal{U}_R$ that approximates any v (blue) in $\mathcal{U}$ with increasing accuracy. The vertical axis represents the, generally multi-dimensional, control input space Ω and the horizontal axis is time.

following:

$$u(t) = w, \qquad \forall t \in [0, 1/R]. \tag{4.0.2}$$

The duration $1/R$ of the primitive input signals is selected for convenience. In general, all that is required is that the duration converges to zero with increasing R. For example, the duration of primitive input signals could alternatively be $1/R^2$ or $1/\log(R)$ with the appropriate adjustments made to the remaining analysis.
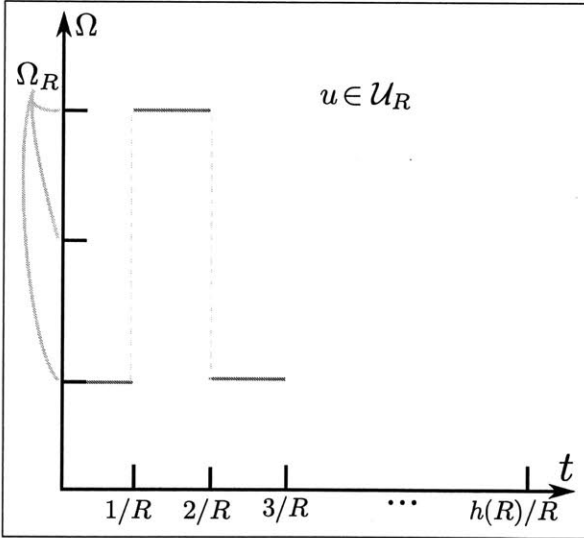
Figure 4-2: Signals from $\mathcal{U}_R$ are piecewise constant taking values in Ω_R on intervals of duration $1/R$. The duration is variable as long as the number of constant segments is less than $h(R)$. The duration of the longest input signal in $\mathcal{U}_R$ is $h(R)/R$ which grows unbounded when $h(R)$ satisfies (4.0.3).

The approximation $\mathcal{U}_R$ is then all possible strings of input signals with length $h(R)$ or less where $h : \mathbb{N} \to \mathbb{N}$ is a horizon or depth limit. That is, an input signal in $\mathcal{U}_R$ will have a time domain $[0, d/R]$ for $d \leq h(R)$, and on each interval $\left[\frac{(i-1)}{R}, \frac{i}{R}\right)$ with $i \in 1, ..., d$, the input signal takes one value from Ω_R. Figure 4-2 illustrates the construction of signals in $\mathcal{U}_R$. The horizon limit can be any function satisfying

$$\lim_{R \to \infty} \frac{R}{h(R)} = 0. \tag{4.0.3}$$

This ensures that with sufficiently high resolution, the time domain of any input signal in $\mathcal{U}$ can be approximated by the time domain of a signal in $\mathcal{U}_R$. Figure 4-3 shows how increasing the resolution improves the available approximations in $\mathcal{U}_R$ to a particular input signal.

Terminology related to $\mathcal{U}_R$: To simplify the discussion of how different input signals relate to one another, it will be useful to introduce some terminology. A

parent of an input signal $w \in \mathcal{U}_R$ with domain $[0, i/R)$ is defined as the input signal $u \in \mathcal{U}_R$ with domain $[0, (i-1)/R)$ such that $w(t) = u(t)$ for all $t \in \left[0, \frac{c \cdot (i-1)}{R}\right)$. In this case, w is a *child* of u. Two signals are *siblings* if they have the same parent. A tree (graph) is defined for each natural number R, with $\mathcal{U}_R$ as the vertex set, and edges defined by ordered pairs of signals (u, w) such that u is the parent of w. To serve as the root of the tree, $\mathcal{U}_R$ is augmented with the special input signal $Id_{\mathcal{U}}$ defined such that $J_{x_0}(Id_{\mathcal{U}}) = 0$ and $\tau(Id_{\mathcal{U}}) = 0$. $Id_{\mathcal{U}}$ has no parent, but is the parent of signals with domain $[0, 1/R]$.

The signal w is an *ancestor* of u if $\tau(w) \le \tau(u)$ and $w(t) = u(t)$ for all $t \in [0, \tau(w))$. In this case u is a *descendant* of w. The *depth* of an input signal in $\mathcal{U}_R$ is the number of ancestors of that input signal.

Remark 1. *With these conventions, each signal is an ancestor, descendant, and sibling of itself.*

4.1 Consistency of the Approximation

Since there is a metric on $\mathcal{U}$, a natural requirement of any approximation of this set by some subset is that for any fixed $u \in \mathcal{U}$,

$$\lim_{R \to \infty} \left(\min_{w \in \mathcal{U}_R} d_{\mathcal{U}}(u, w) \right) = 0. \tag{4.1.1}$$

The interpretation of this condition is as follows: given any fixed[1] input signal u and $\varepsilon > 0$, there is an input signal in $\mathcal{U}_R$ whose distance from u is less than ε with a sufficiently high resolution (i.e. sufficiently large R).

The next result establishes this property for $\mathcal{U}_R$.

Lemma 4.1.1. *For each $u \in \mathcal{U}$ and $\varepsilon > 0$, there exists $R^* > 0$ such that for any $R > R^*$ there exists $w \in \mathcal{U}_R$ satisfying $d_{\mathcal{U}}(u, w) < \varepsilon$.*

Proof. The proof will first rely on Lusin's theorem[2] for the existence of a continuous input signal approximating u. We then approximate the continuous input signal u with an input signal w in $\mathcal{U}_R$.

If follows directly from Lusin's Theorem [Lus12, Fel81] that there exists a continuous input signal $v : [0, \tau(u)] \to \Omega$ such that

$$\mu(\{t \in [0, \tau(u)] : v(t) \ne u(t)\}) < \frac{\varepsilon}{4u_{max}}. \tag{4.1.2}$$

Since u and v have the same time domain, the distance between the two is given by,

$$d_{\mathcal{U}}(u, v) = \int_{[0, \tau(u)]} \|u(t) - v(t)\|_2 \, \mu(dt). \tag{4.1.3}$$

---

[1]Note that this is a weaker condition than requiring the dispersion of $\mathcal{U}_R$ in $\mathcal{U}$ to converge to zero.

[2]Lusin's Theorem is discussed in Appendix C.1.

Then from (4.1.2) this is equal to

$$d_{\mathcal{U}}(u,v) = \int_{\{t \in [0,\tau(u)]:\, v(t) \neq u(t)\}} \|u(t) - v(t)\|_2\, \mu(dt). \qquad (4.1.4)$$

Since Ω is bounded, $\|u(t) - v(t)\|_2 < 2u_{max}$ resulting in the bound

$$d_{\mathcal{U}}(u,v) < \int_{\{t \in [0,\tau(u)]:\, v(t) \neq u(t)\}} 2u_{max}\, \mu(dt) = \frac{\varepsilon}{2}. \qquad (4.1.5)$$

Next, we will construct a w in $\mathcal{U}_R$ such that $d_{\mathcal{U}}(v,w)$ is small. The domain of v is compact so the continuity is also uniform. Let $\delta(\epsilon)$ denote the modulus of continuity. That is,

$$|\sigma - \gamma| < \delta(\epsilon) \Rightarrow \|v(\sigma) - v(\gamma)\|_2 < \epsilon. \qquad (4.1.6)$$

To construct an approximation of v by $w \in \mathcal{U}_R$ choose R sufficiently large so that

1. $\tau(v) < h(R)/R$,

2. $1/R < \min\left\{\frac{\varepsilon}{6u_{max}}, \delta\left(\frac{\varepsilon}{6\tau(v)}\right)\right\}$,

3. there exists an integer $r < h(R)$ such that $0 < \tau(v) - r/R < 1/R$,

4. the dispersion of Ω_R in Ω is less than $\frac{\varepsilon}{6\tau(v)}$.

Note that if these conditions hold for R^*, they hold for all $R > R^*$.

It follows from (2.) and (4.) above, and the uniform continuity of v that for each $i \in \{1, 2, ..., r\}$ and $t \in [(i-1)/R, i/R)$, there exists $\mathrm{v}_i \in \Omega_R$ such that $\|\mathrm{v}_i - v(t)\|_2 < \frac{\varepsilon}{3\tau(v)}$. Select $w \in \mathcal{U}_R$ which is equal to v_i on each of these intervals so that $\tau(w) = r/R$, and $\tau(w) < \tau(v)$. By construction of w we have

$$d_{\mathcal{U}}(v,w) = \int_{[0,r/R]} \|w(t) - v(t)\|_2\, \mu(dt) + u_{max}|r/R - \tau(v)|. \qquad (4.1.7)$$

Then by condition (2.) and (3.),

$$d_{\mathcal{U}}(v,w) < \int_{[0,r/R]} \|w(t) - v(t)\|_2\, \mu(dt) + \frac{\varepsilon}{6}. \qquad (4.1.8)$$

Next, by construction of $w(t)$ we have

$$d_{\mathcal{U}}(v,w) < \int_{[0,r/R]} \frac{\varepsilon}{3\tau(u)}\, \mu(dt) + \frac{\varepsilon}{6}. \qquad (4.1.9)$$

Recall that $r/R < \tau(v)$ so that

$$d_{\mathcal{U}}(v,w) < \int_{[0,\tau(v]} \frac{\varepsilon}{3\tau(u)}\, \mu(dt) + \frac{\varepsilon}{6}. \qquad (4.1.10)$$

40

Integrating the right hand side of the inequality yields

$$d_{\mathcal{U}}(v, w) < \frac{\varepsilon}{3} + \frac{\varepsilon}{6} = \frac{\varepsilon}{2}. \tag{4.1.11}$$

Thus, by the triangle inequality

$$d_{\mathcal{U}}(u, w) \leq d_{\mathcal{U}}(u, v) + d_{\mathcal{U}}(v, w) < \varepsilon. \tag{4.1.12}$$

$\square$

The previous result showed that $\mathcal{U}_R$ consistently approximates $\mathcal{U}$ with respect to the metric $d_{\mathcal{U}}$. However, we are primarily interested in approximating $\mathcal{U}_{\text{goal}}$. Since $\mathcal{U}_{\text{goal}}$ is open with respect to $d_{\mathcal{U}}$, the set $\mathcal{U}_R \cap \mathcal{U}_{\text{goal}}{}^3$ will inherit this property. We will combine this observation with the continuity of $J_{x_{ic}}$ to show that there exist signals in $\mathcal{U}_R \cap \mathcal{U}_{\text{goal}}$ with cost arbitrarily close to the optimal cost:

Theorem 4.1.1. *For any $u \in cl\left(int\left(\mathcal{U}_{\text{goal}}\right)\right)$ and $\varepsilon > 0$ there exists $R^* > 0$ such that for any $R > R^*$*

$$\min_{w \in \mathcal{U}_R \cap \mathcal{U}_{\text{goal}}} \left\{ |J_{x_{ic}}(u) - J_{x_{ic}}(w)| \right\} < \varepsilon. \tag{4.1.13}$$

Proof. Let u be a signal in $cl\left(int\left(\mathcal{U}_{\text{goal}}\right)\right)$. Then every neighborhood of u has a nonempty intersection with $int(\mathcal{U}_{\text{goal}})$. Equivalently, for every $\delta > 0$,

$$B_{\delta/2}(u) \cap int(\mathcal{U}_{\text{goal}}) \neq \emptyset. \tag{4.1.14}$$

Let v be an element of this intersection so that $v \in B_{\delta/2}(u)$ and $v \in int(\mathcal{U}_{\text{goal}})$. Since v is an interior point of $\mathcal{U}_{\text{goal}}$, there exists $\rho > 0$ such that $B_\rho(v)$ is a subset of $int(\mathcal{U}_{\text{goal}})$. Now consider the neighborhood $B_{\min\{\rho, \delta/2\}}(v)$, which is a subset of $int(\mathcal{U}_{\text{goal}})$. By Lemma 4.1.1, there exists an R^* such that for all $R > R^*$, there is a signal u in $\mathcal{U}_R$ satisfying $d_{\mathcal{U}}(w, v) < \min\{\rho, \delta/2\}$ which implies w is in $\mathcal{U}_{\text{goal}}$. Further,

$$\begin{aligned} d_{\mathcal{U}}(w, u) &\leq d_{\mathcal{U}}(w, v) + d_{\mathcal{U}}(v, u) \\ &\leq \min\{\rho, \delta/2\} + \delta/2 \\ &\leq \delta. \end{aligned} \tag{4.1.15}$$

Now by the continuity of $J_{x_{ic}}$, for δ sufficiently small, $d_{\mathcal{U}}(u, v) < \delta$ implies $|J_{x_{ic}}(w) - J_{x_{ic}}(u)| < \varepsilon$ from which the result follows. $\square$

A sufficient condition for every $u \in \mathcal{U}_{\text{goal}}$ to be contained in the closure of the interior of $\mathcal{U}_{\text{goal}}$ is that $\mathcal{U}_{\text{goal}}$ be open which is the case when Assumption A-1 is satisfied.

---

[3]This is the set of signals within the approximation of $\mathcal{U}$ resulting in trajectories that satisfy obstacle avoidance constraints and terminal constraints.

Chapter References

[Fel81] Marcus B Feldman, *A proof of Lusin's theorem*, The American Mathematical Monthly **88** (1981), no. 3, 191–192.

[Lus12] Nikolai Lusin, *Sur les propriétés des fonctions mesurables*, CR Acad. Sci. Paris **154** (1912), 1688–1690.

Chapter 5

The Generalized Label Correcting Method

Equipped with the set $\mathcal{U}_R$, one could enumerate the strings of input signal primitives in $\mathcal{U}_R$ to find the minimum cost input signal in $\mathcal{U}_R \cap \mathcal{U}_{\text{goal}}$. Theorem 4.1.1 tells us that this procedure would solve Problem 2.2.1. However the number of signals in $\mathcal{U}_R$ is $h(R)^{\text{card}(\Omega_R)}$ which grows rapidly as the dispersion of the set of control inputs Ω_R in Ω converges to zero. This type of exhaustive search is analogous to searching over all paths in a graph for a shortest path between two vertices. However, this is an inefficient approach which is remedied with label correcting algorithms.

Label correcting algorithmms: In a conventional *label correcting method*, the algorithm maintains the least cost path p_{label} known to terminate at each vertex of the graph. This path *labels* that vertex. At a particular iteration, if a path p_{new} under consideration does not have lower cost than the path labeling the terminal vertex, the path under consideration is discarded. Justification for this operation is that any extension p_{tail} of the path p_{new} which reaches the goal vertex can be also be used to extend p_{label} to reach the goal vertex. The cost of the extension p_{tail} in both cases will be the same, but the total cost of p_{new} concatenated with p_{tail} will certainly be no less than the total cost of p_{label} concatenated with p_{tail}. Therefore, there is no need to consider any extensions of p_{new}. As a consequence, the subtree of paths originating from the discarded path will not be evaluated.

Generalizing the notion of a label: Observe that the label of a vertex in conventional label correcting algorithms is in fact a label for the paths terminating at that vertex. Then each vertex identifies an equivalence class of paths which terminate at that vertex. Paths within each equivalence class are ordered by their cost, and the efficiency of label correcting methods comes from narrowing the search to minimum cost paths in their associated equivalence class. The generalization is to identify paths associated to trajectories terminating in the same region of the state space instead of the same state. In the context of trajectory planning, this is complicated by the fact that the optimal cost from any particular state to the goal may be discontinuous.

43

This makes it difficult to prove that a trajectory terminating close to another trajectory with lesser cost can be discarded. In this chapter we derive precise conditions for discarding a trajectory that resolves this issue.

5.1 Constructing Equivalence Classes of Signals

The equivalence classes of input signals are induced by a partition of X_{free}. A partition of X_{free} is said to have radius r if each element of the partition is a set contained in a neighborhood of radius r. No further assumptions on the geometry of these sets will be required. A partition of radius r is illustrated in Figure 5-1

For now we only consider hypercube partitions whose radius is controlled by a function $\eta : \mathbb{N} \to \mathbb{R}_{>0}$. For states $p_1, p_2 \in \mathbb{R}^n$ we write $p_1 \overset{R}{\sim} p_2$ if

$$\lfloor \eta(R)p_1 \rfloor = \lfloor \eta(R)p_2 \rfloor, \qquad (5.1.1)$$

where $\lfloor \cdot \rfloor$ is the coordinate-wise floor map (e.g. $\lfloor (2.9, 3.2) \rfloor = (2, 3)$). The equivalence classes of the $\overset{R}{\sim}$ relation define a simple hypercube partition of radius $\sqrt{n}/\eta(R)$. We extend this relation to control inputs by comparing the terminal state of the resulting trajectory. For $u_1, u_2 \in \mathcal{U}_R$ we write $u_1 \overset{\mathcal{U}_R}{\sim} u_2$ if the resulting trajectories terminate in the same hypercube. That is,

Figure 5-1: The green region shows one element of a partition of $\mathbb{R}^2$. This set is contained in a ball of radius r as are all other elements of the partition.

$$u_1 \overset{\mathcal{U}_R}{\sim} u_2 \Leftrightarrow [\varphi_{x_{\text{ic}}}(u_1)](\tau(u_1)) \overset{R}{\sim} [\varphi_{x_{\text{ic}}}(u_2)](\tau(u_2)). \qquad (5.1.2)$$

Figure 5-2 illustrates the intuition behind this equivalence relation.

The principal contribution of this thesis is the *GLC conditions* for ordering input signals among the equivalence classes of the relation $\overset{\mathcal{U}_R}{\sim}$. The GLC conditions define a partial ordering $\prec_R$ on $\mathcal{U}_R$ which is used to identify signals which can be discarded. We write $u_1 \prec_R u_2$ if:

GLC-1 $u_1 \overset{\mathcal{U}_R}{\sim} u_2$,

GLC-2 $\tau(u_1) \leq \tau(u_2)$,

GLC-3 $J_{x_{\text{ic}}}(u_1) \leq J_{x_{\text{ic}}}(u_2)$.

That is, u_1 is $\prec_R$-less than u_2 if they result in trajectories terminating within the same region of the states space (GLC-1), the duration of u_1 is no greater than the duration of u_2 (GLC-2), and the cost of u_1 is less than the cost of u_2 (GLC-3). These conditions are sharper than the ones presented in [PF16] where GLC-3 included a threshold for the difference in cost between u_1 and u_2.

Figure 5-2: Illustration of how a partition of X_{free} induces a partition on $\mathcal{U}$. Input signals u_1, u_2 result in trajectories terminating within the same element of the partition of X_{free}. This is the generalization of identifying paths in a graph terminating at the same vertex as equivalent.

A signal u_1 is called *minimal* if there is no $u_2 \in \mathcal{U}_R$ such that $u_2 \prec_R u_1$. Such a signal can be thought of as being a good candidate for later expansion during the search. Otherwise, it can be discarded. In order for the GLC method to be a resolution complete algorithm, the scaling parameter η must satisfy

$$\lim_{R \to \infty} \frac{R}{L_f \eta(R)} \left(e^{\frac{L_f h(R)}{R}} - 1 \right) = 0. \tag{5.1.3}$$

Figure 5-3 contrasts the trajectories resulting from minimal signals in $\mathcal{U}_R$ with an exhaustive enumeration of $\mathcal{U}_R$ for a two dimensional kinematic point robot. The minimal signals result in approximately optimal trajectories that uniformly cover the free space X_{free} while exhaustively enumerating $\mathcal{U}_R$ requires many more trajectories to be evaluated.

45
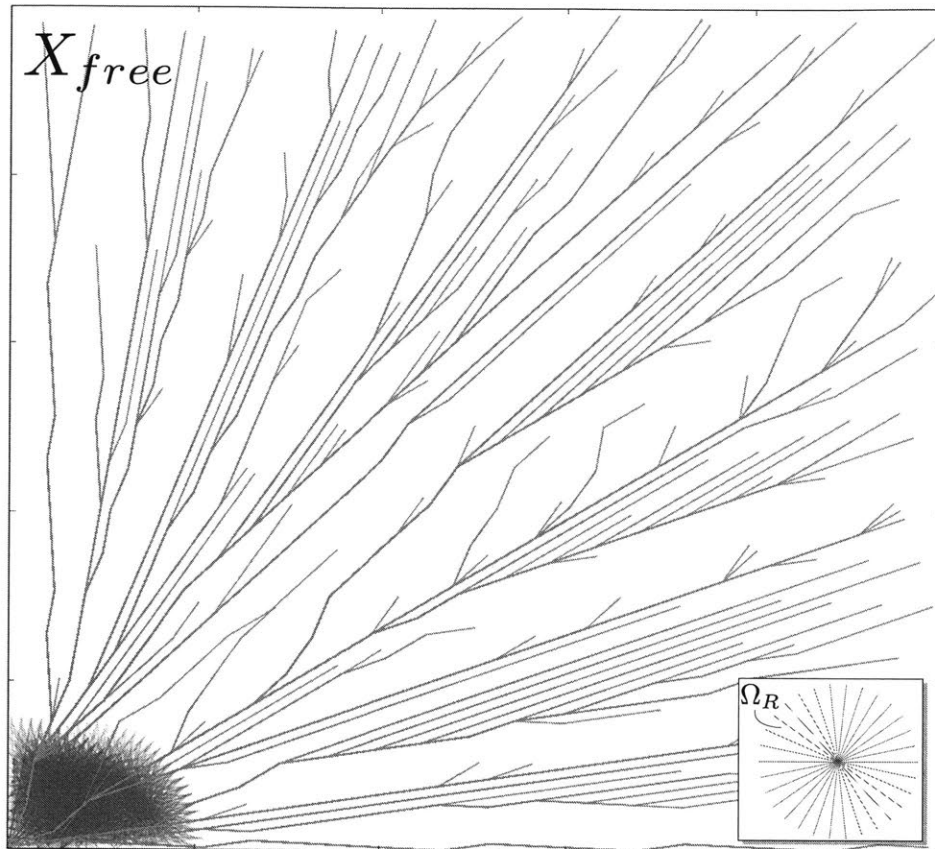
Figure 5-3: The motions resulting from the primitive control inputs Ω_R are shown in the bottom right corner. Executing the first 3000 out of 30^{30} signals from $\mathcal{U}_R$ in uniform cost order (concatenation of 2-3 primitive input signals) results in the blue trajectories illustrated which do not effectively cover X_{free}. On the other hand there are only 417 minimal trajectories in $\mathcal{U}_R$ which cover X_{free} effectively with approximately optimal paths.

5.2 Properties of the GLC Conditions

This section will develop a number of important concepts and basic properties of the partial order $\prec_R$.

The ϵ-interior of the set $\mathcal{X}_{\text{free}}$ and $\mathcal{X}_{\text{goal}}$, are defined by

$$\mathcal{X}_{\text{free}}^{\epsilon} := \{x \in \mathcal{X}_{x_{ic}} : B_{\epsilon}(x) \subset \mathcal{X}_{\text{free}}\},$$
$$\mathcal{X}_{\text{goal}}^{\epsilon} := \{x \in \mathcal{X}_{\text{free}}^{\epsilon} : B_{\epsilon}(x) \subset \mathcal{X}_{\text{goal}}\}. \tag{5.2.1}$$

The inverse image of these sets under the system map $\varphi_{x_{ic}}$ is denoted

$$\mathcal{U}_{\text{free}}^{\epsilon} := \varphi_{x_{ic}}^{-1}(\mathcal{X}_{\text{free}}^{\epsilon}), \qquad \mathcal{U}_{\text{goal}}^{\epsilon} := \varphi_{x_{ic}}^{-1}(\mathcal{X}_{\text{goal}}^{\epsilon}). \tag{5.2.2}$$

Note that if x is an element of $\mathcal{X}_{\text{free}}^{\epsilon}$, then $B_{\epsilon}(x(t)) \subset X_{\text{free}}$. The same is true for $\mathcal{X}_{\text{goal}}$.

To simplify notation we will denote the optimal cost of signals in $\mathcal{U}_{\text{goal}}^{\epsilon} \cap \mathcal{U}_R$ by c_R^{ϵ},

$$c_R^{\epsilon} := \min_{u \in \mathcal{U}_R \cap \mathcal{U}_{\text{goal}}^{\epsilon}} \{J_{x_{ic}}(u)\}. \tag{5.2.3}$$

Like, the optimal cost c^* and the approximate optimal cost c_R, c_R^{ϵ} can, in some cases, be ∞. An intuitive, but important property regarding the cost c_R^{ϵ} is that it converges to the optimal cost c^* as ε tends to 0 and R tends to ∞.

Lemma 5.2.1. *If* $\lim_{R \to \infty} \epsilon(R) = 0$, *then* $\lim_{R \to \infty} c_R^{\epsilon(R)} = c^*$.

Proof. Since the infimum in (2.2.5) may not be attained, the first step is to identify an input signal which is arbitrarily close to the optimal cost. By the definition of c^* in (2.2.5), for any $\varepsilon > 0$ there exists $\omega \in \mathcal{U}_{\text{goal}}$ such that $J_{x_{ic}}(\omega) - \varepsilon/2 < c^*$.

Next, we use the topological properties discussed in Chapter 3 to construct a neighborhood of ω containing signals in $\mathcal{U}_{\text{goal}}$. Since $\mathcal{U}_{\text{goal}}$ is open and $\varphi_{x_{ic}}$ is continuous, there exists $\tilde{r} > 0$ and $\rho > 0$ such that $B_{\tilde{r}}(\omega) \subset \mathcal{U}_{\text{goal}}$ and $\varphi_{x_{ic}}(B_{\tilde{r}}(\omega)) \subset B_{\rho}(\varphi_{x_{ic}}(\omega))$. Thus, $\omega \in \mathcal{U}_{\text{goal}}^{\rho}$. From the continuity of J in Lemma 3.3.3 there also exists a positive $\delta < r$ such that for any signal v with $d_{\mathcal{U}}(v, \omega) < \delta$ we have $|J_{x_{ic}}(\omega) - J_{x_{ic}}(v)| < \varepsilon/2$.

The next step is to find a feasible signal in $\mathcal{U}_R$ which closely approximates v. Choose R^* to be sufficiently large such that $R > R^*$ implies $\epsilon(R) < \rho$ and $B_{\delta}(\omega) \cap \mathcal{U}_R \neq \emptyset$. Such a resolution R^* exists by Theorem 4.1.1 and the assumption $\lim_{R \to \infty} \epsilon(R) = 0$. Now choose $u \in B_{\delta}(\omega) \cap \mathcal{U}_R$. Then $|J_{x_{ic}}(u) - J_{x_{ic}}(\omega)| < \varepsilon/2$ and $u \in \mathcal{U}_{\text{goal}}^{\rho} \subset \mathcal{U}_{\text{goal}}^{\epsilon(R)}$.

Finally, we use the triangle inequality to show that u has nearly the optimal cost. Then, by definition of $c_R^{\epsilon(R)}$, $u \in \mathcal{U}_{\text{goal}}^{\epsilon(R)}$ implies $c_R^{\epsilon(R)} \leq J_{x_{ic}}(u)$. Finally, by the triangle inequality,

$$|J_{x_{ic}}(u) - c^*| < |J_{x_{ic}}(u) - J_{x_{ic}}(\omega)| + |J_{x_{ic}}(\omega) - c^*| < \varepsilon. \tag{5.2.4}$$

Rearranging the expression yields $J_{x_{ic}}(u) < c^* + \varepsilon$ and thus, $c_R^{\epsilon(R)} < c^* + \varepsilon$. The result follows since the choice of ε is arbitrary. $\qquad \square$

Some new notation: At this point it will be advantageous to introduce a concatenation operation on elements of $\mathcal{U}$ and $\mathcal{X}_{x_0}$. For $s_1, s_2 \in \mathcal{U} \cup \mathcal{X}_{x_0}$, their concatenation

$s_1 s_2$ is defined by

$$[s_1 s_2](t) := \begin{cases} s_1(t), \ t \in [0, \tau(s_1)) \\ s_2(t - \tau(s_1)), \ t \in [\tau(s_1), \tau(s_1) + \tau(s_2)] \end{cases} . \tag{5.2.5}$$

The concatenation is illustrated in Figure 5-4. The concatenation operation will be
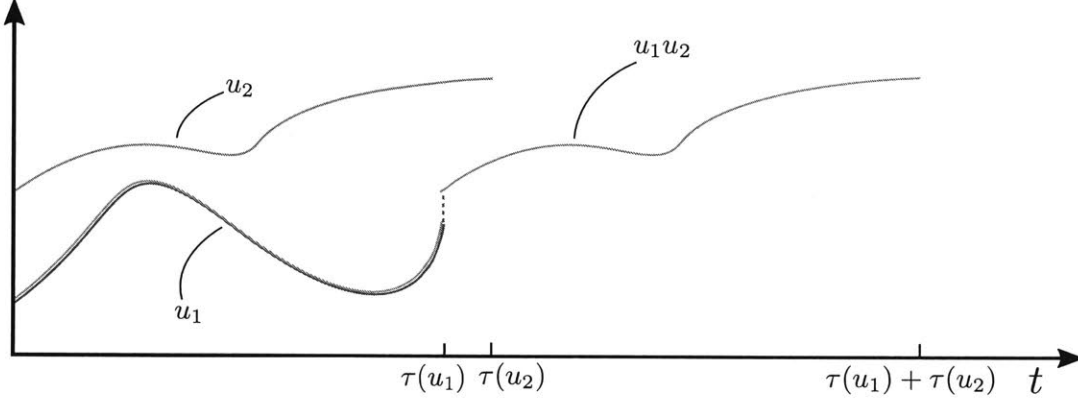

Figure 5-4: Signals u_1 and u_2 are illustrated in blue and green respectively. Their concatenation $u_1 u_2$ is illustrated in red.

useful together with the following equalities for $u_1, u_2 \in \mathcal{U}$,

$$J_{x_0}(u_1 u_2) = J_{x_0}(u_1) + J_{[\varphi_{x_0}(u_1)](\tau(u_1))}(u_2) \tag{5.2.6}$$

$$\varphi_{x_0}(u_1 u_2) = \varphi_{x_0}(u_1) \varphi_{[\varphi_{x_0}(u_1)](\tau(u_1))}(u_2). \tag{5.2.7}$$

Equations (5.2.6) and (5.2.7) follow directly from (2.1.3). The interpretation of (5.2.6) is that the cost of u_1 concatenated with u_2 is equal to the cost of executing the control signal u_1 from x_0 plus the cost of executing u_2 from the terminal state of the trajectory resulting from u_1.

The next result can be interpreted as a generalization of Bellman's *principle of optimality* [Bel56] and is the basis for the generalized label correcting method in the same way that the principle of optimality is the basis for label correcting methods. Figure 5-5 illustrates the statement of the next theorem.

Theorem 5.2.1 (Principle of Optimality). *Let* $\gamma = \frac{\sqrt{n}}{\eta(R)} \frac{L_g}{L_f} \left(e^{L_f \left(\frac{h(R)}{R} - \tau(u_j) \right)} - 1 \right)$, *and* $\delta = \frac{\sqrt{n}}{\eta(R)} e^{L_f \left(\frac{h(R)}{R} - \tau(u_j) \right)}$. *If* $u_i, u_j \in \mathcal{U}_R \cap \mathcal{U}_{\text{free}}$ *and satisfy* $u_i \prec_R u_j$, *then for each descendant of* u_j *in* $\mathcal{U}_R \cap \mathcal{U}_{\text{goal}}^\delta$ *with cost* c_j, *there exists a descendant of* u_i *in* $\mathcal{U}_R \cap \mathcal{U}_{\text{goal}}$ *with cost* $c_i \leq c_j + \gamma$.

Proof. Suppose there exists $w \in \mathcal{U}_R$ such that $u_j w \in \mathcal{U}_R \cap \mathcal{U}^\delta$. By GLC-1 the signals satisfy $u_i \overset{\mathcal{U}_R}{\sim} u_j$ which means

$$\| [\varphi_{x_{ic}}(u_i w)](\tau(u_i)) - [\varphi_{x_{ic}}(u_j w)](\tau(u_j)) \|_2 \leq \frac{\sqrt{n}}{\eta(R)}. \tag{5.2.8}$$

48

Note that $\tau(w) \le h(R)/R - \tau(u_j)$ since $u_j w$ has depth no greater than $h(R)$. Then, by Lemma (3.2.1), for all $t \in [0, \tau(w)]$,

$$\|[\varphi_{x_{\mathrm{ic}}}(u_i w)](t + \tau(u_i)) - [\varphi_{x_{\mathrm{ic}}}(u_j w)](t + \tau(u_j))\| \le \frac{\sqrt{n}}{\eta(R)} e^{L_f\left(\frac{h(R)}{R} - \tau(u_j)\right)}$$
$$= \delta. \tag{5.2.9}$$

Thus, $u_i w \in \mathcal{U}_{\mathrm{goal}}$. Next, we show that $c_i \le c_j + \gamma$. From equation (5.2.6),

$$J_{x_{\mathrm{ic}}}(u_i w) = J_{x_{\mathrm{ic}}}(u_i) + J_{[\varphi_{x_{\mathrm{ic}}}(u_i)](\tau(u_i))}(w). \tag{5.2.10}$$

Combining the continuity of the cost functional with respect to the initial condition (Lemma 3.3.4) and equation (5.2.8) yields

$$\left| J_{[\varphi_{x_{\mathrm{ic}}}(u_i)](\tau(u_i))}(w) - J_{[\varphi_{x_{\mathrm{ic}}}(u_j)](\tau(u_j))}(w) \right|$$
$$\le \|[\varphi_{x_{\mathrm{ic}}}(u_i)](\tau(u_i)) - [\varphi_{x_{\mathrm{ic}}}(u_j)](\tau(u_j))\|_2 \cdot \frac{L_g}{L_f} \left(e^{L_f\left(\frac{h(R)}{R} - \tau(u_j)\right)} - 1 \right)$$
$$\le \frac{\sqrt{n}}{\eta(R)} \frac{L_g}{L_f} \left(e^{L_f\left(\frac{h(R)}{R} - \tau(u_j)\right)} - 1 \right)$$
$$= \gamma. \tag{5.2.11}$$

Then combining (5.2.10) and (5.2.11) yields

$$J_{x_{\mathrm{ic}}}(u_i w) \le J_{x_{\mathrm{ic}}}(u_j) + J_{[\varphi_{x_{\mathrm{ic}}}(u_j)](\tau(u_j))}(w) + \left| J_{[\varphi_{x_{\mathrm{ic}}}(u_i)](\tau(u_i))}(w) - J_{[\varphi_{x_{\mathrm{ic}}}(u_j)](\tau(u_j))}(w) \right|$$
$$\le J_{x_{\mathrm{ic}}}(u_j w) + \frac{\sqrt{n}}{\eta(R)} \frac{L_g}{L_f} \left(e^{L_f\left(\frac{h(R)}{R} - \tau(u_j)\right)} - 1 \right).$$
$$\tag{5.2.12}$$

Therefore $c_i \le c_j + \gamma$. $\qquad\square$

In reference to the above theorem, since $u_j w \in \mathcal{U}_R$ which is limited to signals with depth $h(R)$, the signal w, which is concatenated with u_i to form a signal in $\mathcal{U}_{\mathrm{goal}} \cap \mathcal{U}_R$, necessarily satisfies

$$\mathtt{depth}(w) \le h(R) - \mathtt{depth}(u_j), \tag{5.2.13}$$

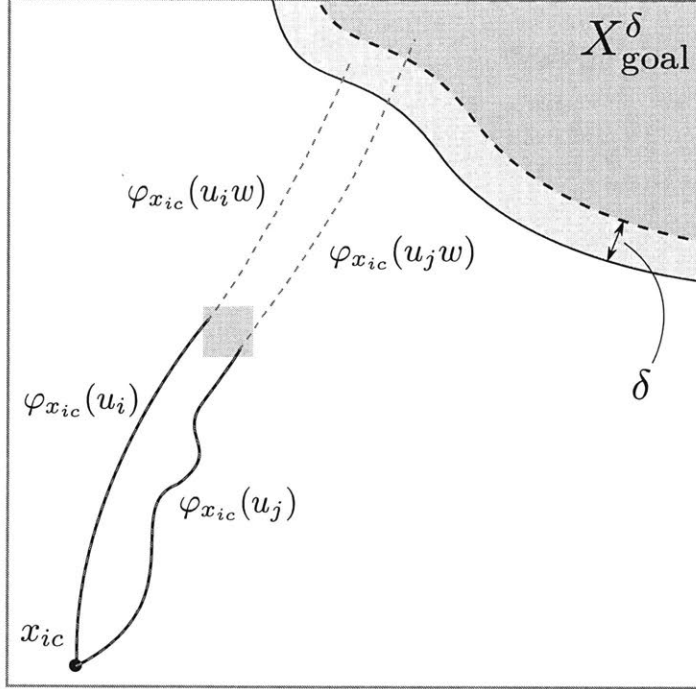where $\mathtt{depth}$ is the depth of a particular signal.

Figure 5-5: Illustration of the statement of Theorem 5.2.1. The input signals u_i and u_j result in trajectories (black curves) terminating in the same partition element (grey square). If $u_i \prec_R u_j$, then for any descendant $u_j w$ of u_j, the descendant $u_i w$ of u_i will have lesser cost, and will result in a trajectory terminating near the trajectory resulting from $u_j w$.

5.3   A GLC Algorithm

Pseudocode for a GLC algorithm is defined in Algorithm 1 below. As the name suggests, it is very similar to a canonical label correcting algorithm. A set Q serves as a priority queue of candidate signals. A set Σ contains signals representing labels of $\overset{\mathcal{U}_R}{\sim}$ equivalence classes.

The method $\texttt{expand}(u)$ returns the set of all children of u. The method $\texttt{pop}(Q)$ deletes from Q, and returns an input signal $\hat{u}$ such that

$$\hat{u} \in \underset{u \in Q}{\operatorname{argmin}} \left\{ J_{x_{\mathrm{ic}}}(u) \right\}, \tag{5.3.1}$$

so that the presented GLC algorithm is a best-first search[1]. The method $\texttt{find}(u, \Sigma)$ returns $w \in \Sigma$ such that $u \overset{R}{\sim} w$ or $\texttt{NULL}$ if no such w is present in Σ. Problem specific collision and goal checking subroutines are used to evaluate $u \in \mathcal{U}_{\mathrm{feas}}$ and $u \in \mathcal{U}_{\mathrm{goal}}$. The method $\texttt{depth}(u)$ returns the number of ancestors of u.

The algorithm begins by adding the root $Id_{\mathcal{U}}$ to the queue (line 1), and then enters a loop which recursively removes and expands the top of the queue (line 3)

---

[1]Like canonical label correcting methods, there are many variations that utilize alternative orderings. For example, the addition of an *admissible heuristic* [HNR68] in (5.3.1) can be used to guide the search without affecting the solution accuracy. Admissible heuristics for kinodynamic motion planning are discussed in detail in Chapter 6.

Algorithm 1 Generalized Label Correcting (GLC) Method

1: $Q \leftarrow \{Id_\mathcal{U}\}$, $\Sigma \leftarrow \emptyset$, $S \leftarrow \emptyset$
2: **while** $Q \neq \emptyset$
3:     $u \leftarrow \mathrm{pop}(Q)$
4:     **if** $u \in \mathcal{U}_{\mathrm{goal}}$
5:        **return** $(J_{x_{ic}}(u), u)$
6:     $S \leftarrow \mathrm{expand}(u)$
7:     **for** $w \in S$
8:        $z \leftarrow \mathrm{find}(w, \Sigma)$
9:        **if** $(w \notin \mathcal{U}_{feas.} \vee (z \prec_R w) \vee \mathrm{depth}(w) \geq h(R))$
10:           $S \leftarrow S \setminus \{w\}$
11:        **else if** $J_{x_{ic}}(w) < J_{x_{ic}}(z)$
12:           $\Sigma \leftarrow (\Sigma \setminus \{z\}) \cup \{w\}$
13:     $Q \leftarrow Q \cup S$
14: **return** (∞, NULL)

---

adding children to S (line 4). If the queue is empty the algorithm terminates (line 2) returning NULL (line 14). Each signal in S (line 5) is checked for membership in $\mathcal{U}_{\mathrm{goal}}$ in which case the algorithm terminates returning a feasible solution with approximately the optimal cost. Otherwise, the signals are checked for infeasibility or suboptimality by the GLC conditions (line 9). Next, a relabeling condition for the associated equivalence classes (i.e. grid cells) of remaining signals is checked (line 11). Finally, remaining signals in S are added to the queue (line 13).

In addition to the problem data, the algorithm requires two functions $h(R)$ and $\eta(R)$ satisfying equations (4.0.3) and (5.1.3), and the control inputs Ω_R satisfying (4.0.1).

5.4   Proof of Resolution Completeness

The goal of this section is to prove that Algorithm 1 is resolution complete for Problem 2.2.1. Algorithm 1 is very similar to classical label correcting algorithms and the proof parallels standard proofs of completeness for these methods. The reader may benefit from reviewing the proof in the simpler case where the algorithm is applied to a shortest path problem in a graph (e.g. [Ber95, chapter 2] or Appendix D).

In Theorem 5.2.1, the quantity δ was constructed based on: the radius of the partition of $\mathcal{X}_{\mathrm{free}}$, the sensitivity of solutions to initial conditions determined by Lemma 3.2.1, and the maximum duration of signals in $\mathcal{U}_R$. In the proof of the next result, we will recursively apply Theorem 5.2.1 and it will be useful to define the following sum that arises when repeatedly applying Theorem 5.2.1,

$$\delta_k := \sum_{i=0}^{k} \frac{\sqrt{n}}{L_f \eta(R)} e^{\frac{L_f(h(R)-i)}{R}}. \tag{5.4.1}$$

51

When $k = h(R)$, the sum satisfies the inequality

$$\sum_{i=0}^{k} \frac{\sqrt{n}}{L_f \eta(R)} e^{\frac{L_f(h(R)-i)}{R}} \leq \frac{R\sqrt{n}}{L_f \eta(R)} \left(e^{L_f h(R)/R} - 1\right), \tag{5.4.2}$$

which is derived in Appendix E.3. Similarly, it will be convenient to define the quantity γ_k related to γ of Theorem 5.2.1,

$$\gamma_k := \sum_{i=0}^{k} \frac{\sqrt{n}}{\eta(R)} \frac{L_g}{L_f} \left(e^{L_f\left(\frac{h(R)}{R}-i\right)} - 1\right), \tag{5.4.3}$$

which is bounded by

$$\sum_{i=0}^{h(R)} \frac{\sqrt{n}}{\eta(R)} \frac{L_g}{L_f} \left(e^{L_f\left(\frac{h(R)}{R}-i\right)} - 1\right) \leq \frac{R\sqrt{n}}{\eta(R)} \frac{L_g}{L_f} \left(e^{L_f h(R)/R} - \frac{h(R)}{R}\right). \tag{5.4.4}$$

An important property of the right hand side of (5.4.2) and (5.4.4) is that they converge to zero when $\eta(R)$ satisfies equation (5.1.3). Thus, $\delta_{h(R)}$ and $\gamma_{h(R)}$ converge to zero as R tends to infinity.

The pruning operation in lines 9-10 of Algorithm 1, in a sense, discards candidate input signals as liberally as possible. In the next theorem, it is shown that a signal in $\mathcal{U}_{\text{goal}}^{\delta_{h(R)}}$ with cost less than $c_R^{\delta_{h(R)}} + \gamma_{h(R)}$ is eventually evaluated by the algorithm in line 4 despite this pruning operation.

Theorem 5.4.1. *The* GLC *method described by Algorithm 1 terminates in finite time and returns a solution with cost less than or equal to $c_R^{\delta_{h(R)}} + \gamma_{h(R)}$.*

Proof. (Finite running time) The queue is a subset of $\mathcal{U}_R \cup \{Id_{\mathcal{U}}\}$ and at line 3 in each iteration a lowest cost signal u is removed from the queue. In line 13, only children of the current signal u are added to the queue. Since $\mathcal{U}_R$ is organized as a tree and has no cycles, any signal u will enter the queue at most once. Therefore the queue must be empty after a finite number of iterations so the algorithm terminates.

(Approximate Optimality) Next, consider as a point of contradiction the hypothesis that the output has cost greater than $c_R^{\delta_{h(R)}} + \gamma_{h(R)}$. Then it is necessary that $c_R^{\delta_{h(R)}} + \gamma_{h(R)} < \infty$, and by the definition of $c_R^{\delta_{h(R)}}$ in (5.2.3), it is also necessary that $\mathcal{U}_R \cap \mathcal{U}_{\text{goal}}^{\delta_{h(R)}}$ is non-empty.

Choose $u^* \in \mathcal{U}_R \cap \mathcal{U}_{\text{goal}}^{\delta_{h(R)}}$ with cost $J_{x_{\text{ic}}}(u^*) = c_R^{\delta_{h(R)}}$. It follows from the contradiction hypothesis that u^* does not enter the queue. Otherwise, by (5.3.1) it would be evaluated before any signal of cost greater than $c_R^{\delta_{h(R)}}$ and the algorithm would terminate returning this input signal. If u^* does not enter the queue, then a signal u_0 must at some iteration be present in Σ which prunes an ancestor a_0 of u^* ($u_0 \prec_R a_0$ in line 9). This ancestor must satisfy $\text{depth}(a_0) > 0$ since the ancestor with depth 0 is $Id_{\mathcal{U}}$ which enters queue in line 1. By Theorem 5.2.1, u_0 has a descendant of the form

52

$u_0 d_0 \in \mathcal{U}_{\text{goal}}^{\delta_{h(R)} - \delta_0}$ and $J_{x_{\text{ic}}}(u_0 d_0) \leq c_R^{\delta_{h(R)}} + \gamma_0$. Additionally, $\texttt{depth}(a_0) > 0$ implies $\texttt{depth}(d_0) \leq h(R) - 1$ by equation (5.2.13).

Having pruned u^* in lines 9-10, the signal $u_0 \in \Sigma$, or a sibling which prunes u_0 (and by the transitivity of $\prec_R$, prunes u^*) must at some point be present in the queue (cf. line 12-13). Of these two, denote the one that ends up in the queue by $\tilde{u}_0$. Since $\tilde{u}_0$ is at some point present in the queue and $\tilde{u}_0 d_0 \in \mathcal{U}_{\text{goal}}^{\delta_{h(R)} - \delta_0}$, a signal $u_1 \in \Sigma$ must prune an ancestor a_1 of $\tilde{u}_0 d_0$ ($u_1 \prec_R a_1$ in lines 9-10). Since $\tilde{u}_0$ is at some point present in the queue, the ancestor a_1 of $\tilde{u}_0 d_0$, must have greater depth than $\tilde{u}_0$. By Theorem 5.2.1, u_1 has a descendant of the form $u_1 d_1 \in \mathcal{U}_{\text{goal}}^{\delta_{h(R)} - \delta_1}$ and $J_{x_{\text{ic}}}(u_1 d_1) \leq c_R^{\delta_{h(R)}} + \gamma_1$. Additionally, $\texttt{depth}(a_1) > \tilde{u}_0$ implies $\texttt{depth}(d_1) \leq \texttt{depth}(d_0) - 1 \leq h(R) - 2$ by equation (5.2.13).

Continuing this line of deduction leads to the observation that a signal $u_{h(R)-1}$, with a descendant of the form $u_{h(R)-1} d_{h(R)-1} \in \mathcal{U}_{\text{goal}}^{\delta_{h(R)} - \delta_{h(R)-1}}$ and $J_{x_{\text{ic}}}(u_{h(R)-1} d_{h(R)-1}) \leq c_R^{\delta_{h(R)}} + \gamma_{h(R)-1}$, will be present in the queue; and $\texttt{depth}(d_{h(R)-1}) \leq 1$. Since $u_{h(R)-1}$ is at some point present in the queue, a signal $u_{h(R)} \in \Sigma$ must prune an ancestor $a_{h(R)}$ of $u_{h(R)-1} d_{h(R)-1}$ ($u_{h(R)} \prec_R a_{h(R)}$ in lines 9-10). Since $u_{h(R)-1}$ is at some point present in the queue, the ancestor $a_{h(R)}$ of $u_{h(R)-1} d_{h(R)-1}$, must have greater depth than $u_{h(R)-1}$, and therefore, is equal to $u_{h(R)-1} d_{h(R)-1}$. Thus, $u_{h(R)} \in \mathcal{U}_{\text{goal}}^{\delta_{h(R)} - \delta_{h(R)-1}}$ and $J(u_{h(R)}) \leq c_R^{\delta_{h(R)}} + \gamma_{h(R)}$. Then $u_{h(R)}$ or a sibling which prunes $u_{h(R)}$ will be added to the queue; a contradiction of the hypothesis since this signal will be removed from the queue and the algorithm will terminate, returning this signal in line 7. $\qquad \square$

The choice of $\delta_{h(R)}$ and $\gamma_{h(R)}$ in Theorem 5.4.1 converge to zero as R tends to infinity by (5.1.3). Then by Lemma 5.2.1 we have $c_R^{h(R)} + \gamma_{h(R)} \to c^*$. An immediate corollary is that the GLC method is resolution complete. It is important to note that for low resolution R, it is possible that $c_R^{h(R)} = \infty$.

Corollary 5.4.1. *Let w_R be the signal returned by the* GLC *method for resolution R. Then $\lim_{R \to \infty} J_{x_{\text{ic}}}(w_R) = c^*$. That is, the* GLC *method is a resolution complete algorithm for the optimal kinodynamic motion planning problem.*

5.5 Kinodynamic Motion Planning Examples

The GLC method (Algorithm 1) was tested on five problems and compared, when applicable, to the implementation of RRT* from [Kar] and SST from [LLB]. The goal is to examine the performance of the GLC method on a wide variety of problems. The examples include under-actuated nonlinear systems, multiple cost objectives, and environments with/without obstacles. Note that adding obstacles effectively speeds up the GLC method since it reduces the size of the search tree.

Another focus of the examples is on real-time application. In each example the running time for GLC method to produce a (visually) acceptable trajectory is comparable to the execution time. Of course this will vary with problem data and computing hardware.

Implementation Details: The GLC method was implemented in C++ and run with a 3.70GHz Intel Xeon CPU. The set Q was implemented with an STL priority queue so that the $\text{pop}(Q)$ method and insertion operations have logarithmic complexity in the size of Q. The set Σ was implemented with an STL set which uses a binary search tree so that $\text{find}(w, \Sigma)$ also has logarithmic complexity in the size of Σ.

Sets X_{free} and X_{goal} are described by algebraic inequalities. The approximation of the input space Ω_R is constructed by uniform deterministic sampling of R^m (i.e. the resolution R raised to the power m which is the dimension of the input space) controls from Ω. Recall m is the dimension of the input space.

Evaluation of $u \in \mathcal{U}_{\text{feas}}$ and $u \in \mathcal{U}_{\text{goal}}$ is approximated by first numerically computing $\varphi_{x_{\text{ic}}}(u)$ with Euler integration (except for RRT* which uniformly samples along the local planning solution). The number of time-steps is given by $N = \lceil \tau(u)/\Delta \rceil$ with duration $\tau(u)/N$. Maximum time-steps Δ are 0.005 for the first problem, 0.1 for the second through fourth problem, and 0.02 for the last problem. Feasibility is then approximated by collision checking at each time-step along the trajectory.

An additional scaling of the input signal primitive duration by γ is introduced in these examples. Theoretically, the output of the algorithm for any value of γ will be consistent with Theorem 5.4.1, but in practice it is advantageous to select a scaling γ representing a characteristic time-scale for the problem. For example, if the motion is expected to take on the order of microseconds, then selecting $\gamma = 10^{-6}$ would adjust the time-scale of the search appropriately.

5.5.1 Shortest Path Problem

A shortest path problem in $\mathbb{R}^n$ can be represented by the dynamic model

$$\frac{d}{dt}x(t) = u(t), \tag{5.5.1}$$

the running-cost

$$g(x, u) = 1, \tag{5.5.2}$$

and the control input space

$$\Omega = \{u \in \mathbb{R}^2 : \|u\|_2 = 1\}. \tag{5.5.3}$$

In this example, X_{free} and X_{goal} are described by polyhedral sets illustrated in Figure 5-6.

The parameters for the GLC method are summarized in Table 5.1. Note that $h(R)$ and $\eta(R)$ satisfy the asymptotic constraints (4.0.3) and (5.1.3). The GLC, SST, and RRT* methods were all tested in this example. The average performance from 10 trials is reported for the SST and RRT* methods. Figure 5-6 summarizes the results. This example also considers a guided search using the cost of the RRT* local planning subroutine solution to the goal as a heuristic.

This is the only example in this section where the exact solution to the problem is known. In this case we can compare the relative convergence rates of the GLC

Table 5.1: Tuning parameter selection for the shortest path example.

Resolution range: R	$R = \{20, 25, ..., 200\}$
Horizon limit: $h(R)$	$100 R \log(R)$
Partition scaling: $\eta(R)$	$R^2/300$
Control primitive duration: γ/R	$10/R$

method, SST, and RRT*.

5.5.2 Torque-Limited Pendulum Swing-Up

A minimum-time torque-limited pendulum swing-up problem can be represented by the dynamic model

$$\dot{\theta}(t) = \omega(t), \qquad \dot{\omega}(t) = -\sin(\theta(t)) + u(t), \tag{5.5.4}$$

with the running cost

$$g(\theta, \omega, u) = 1, \tag{5.5.5}$$

and the control input space

$$\Omega = [-0.2, 0.2]. \tag{5.5.6}$$

The free space and goal set are described by

$$X_{\text{free}} = \mathbb{R}^2, \qquad X_{\text{goal}} = \{(\theta(t), \omega(t)) \in \mathbb{R}^2 : \|(\theta(t) \pm \pi, \omega(t)\|_2 \leq 0.1\}. \tag{5.5.7}$$

The initial state is the origin $(\theta(0), \omega(0)) = (0, 0)$.

The parameters for the GLC method are summarized in Table 5.1. The GLC and

Table 5.2: Tuning parameter selection for the pendulum swing-up example.

Resolution range: R	$R = \{4, 5, 6, 7, 8\}$
Horizon limit: $h(R)$	$100 R \log(R)$
Partition scaling: $\eta(R)$	$R^{5/2}/16$
Control primitive duration: γ/R	$6/R$

SST methods were tested in this example with the average performance from 10 trials reported for the SST method. Figure 5-7 summarizes the results.

5.5.3 Torque Limited Acrobot Swing-Up

The acrobot is a double link pendulum actuated at the middle joint. The expression for the four dimensional system dynamics are cumbersome to describe and we refer to [Spo95] for the details. The model parameters, free space, and goal region are identical to those in the benchmark provided in [LLB] with the exception that the

radius of the goal region is reduced from 2.0 to 0.5. A minimum-time running-cost

$$g(x, u) = 1, \tag{5.5.8}$$

is used. The control input space is $\Omega = [-4.0, 4.0]$ representing the minimum and maximum torque that can be applied at the middle joint.

The parameters for the GLC method are summarized in Table 5.3. The GLC and

Table 5.3: Tuning parameter selection for the acrobot swing-up example.

Resolution range:	$R \in \{4, 5, ..., 10\}$
Horizon limit: $h(R)$	$100R \log(R)$
Partition scaling: $\eta(R)$	$R^2/16$
Control primitive duration: γ/R	$6/R$

SST methods were tested in this example with the average performance from 10 trials reported for the SST method. Figure 5-8 summarizes the results.

5.5.4 Thrust Limited 3D Point Robot

To emulates the mobility of an agile aerial vehicle (e.g. a quadrotor with high bandwidth attitude control), the system dynamics are modeled by

$$\dot{x}(t) = v(t), \qquad \dot{v}(t) = 5.0 \cdot u(t) - 0.1 \cdot v(t) \cdot \|v(t)\|_2, \tag{5.5.9}$$

where $x(t)$, $v(t)$, and $u(t)$ are each elements of $\mathbb{R}^3$; there are a total of six states and three control inputs. The quadratic dissipative force anti-parallel to the velocity $v(t)$ models aerodynamic drag during high speed flight. The minimum time running cost is

$$g(x, v, u) = 1, \tag{5.5.10}$$

and the input space is

$$\Omega = \{u \in \mathbb{R}^3 : \|u\|_2 \le 1\}. \tag{5.5.11}$$

The planning task is for the point robot to navigate from an initial state in one room to a destination in an adjacent room connected by a small window. This free space is illustrated in Figure 5-9. The dark blue marker in the leftmost room illustrates the goal configuration, and the light blue marker in the rightmost room illustrates the starting configuration. The velocity is initially zero, and the terminal velocity is left free. A guided search is also considered with heuristic given by the distance to the goal divided by the maximum speed v of the robot (A maximum speed of $\sqrt{50}$ can be determined from the dynamics and input constraints). The parameters for the GLC method are summarized in Table 5.1.

The control input approximation Ω_R is generated by an increasing number of points distributed uniformly on Ω. However, because of the symmetry of the sphere, the computational procedure for generating Ω_R in this case produces the same set of points up to a random orthogonal transformation [PF17]. Since there is some

Table 5.4: Tuning parameter selection for the thrust limited 3D point robot example.

Resolution range:	$R \in \{8, 9, ..., 12\}$
Horizon limit: $h(R)$	$100R\log(R)$
Partition scaling: $\eta(R)$	$R^2/64$
Control primitive duration: γ/R	$10/R$

randomness in this example, the results reported in Figure 5-10 are averaged over 10 trials both for the GLC and SST algorithm.

Figure 5-9 shows the three outputs of the GLC algorithm at three different resolutions to illustrate the improvement in solution quality with increasing resolution.

5.5.5 Nonholonomic Wheeled Robot

The dynamic model emulating the mobility of a wheeled robot is given by

$$\dot{p}_1(t) = \cos(\theta(t)), \quad \dot{p}_2(t) = \sin(\theta(t)), \quad \dot{\theta}(t) = u(t). \tag{5.5.12}$$

A running cost which penalizes a combination of time and lateral acceleration (believed to be correlated with rider comfort) is given by

$$g(p_1, p_2, \theta, u) = 1 + 2u^2. \tag{5.5.13}$$

The input space, which is consistent with a turning radius of 1, is

$$\Omega = [-1, 1]. \tag{5.5.14}$$

The parameters for the GLC method used in this example are summarized in Table 5.1.

Table 5.5: Tuning parameter selection for the wheeled robot example.

Resolution range:	$R \in \{4, 5, ..., 9\}$
Horizon limit: $h(R)$	$5R\log(R)$
Partition scaling: $\eta(R)$	$R^{5/\pi}/15$
Control primitive duration: γ/R	$10/R$

Figure 5-11 summarizes the performance of the GLC algorithm on this problem.

5.5.6 Observations and Discussion

The first observation is that all algorithms tested generally produce lower cost solutions with increased run-time. In example 5.5.1, where the optimal cost was known a-priori, the GLC and RRT* algorithms were observed to converge to this value, and the SST algorithm converged to some approximately optimal cost. In the remaining

examples, where the optimal cost was unknown and RRT* was not applicable, the GLC and SST algorithms were observed to produce solutions of generally decreasing cost with increased run-time. The local planning subroutine for the RRT* algorithm in example 5.5.1 was simply the line segment connecting two points. This is requires virtually no computation to generate making RRT* the preferred algorithm.

Another observation was that the cost of solutions output by the GLC algorithm generally decreased with increasing resolution, but did not decrease monotonically. This is not inconsistent with Corollary 5.4.1 since the result only claims convergence. The cause of the nonmonotonic convergence is that each run of Algorithm 1 operates on $\mathcal{U}_R$ for a fixed R, and since $\mathcal{U}_R \not\subset \mathcal{U}_{R+1}$ it is possible that an optimal signal in $\mathcal{U}_R$ may be better than any signal in $\mathcal{U}_{R+1}$ for certain values of R. In example 5.5.4, Ω_R was randomly generated which required averaging several trials. This had a smoothing effect on the performance curves which is analogous to the effect of averaging the outputs of RRT* and SST.

A very important distinction between the GLC and the other methods RRT* and SST is that the GLC algorithm must run to completion before a solution is returned, while RRT* and SST are incremental algorithms that can be interrupted at any time and return the current best solution. This is a desirable feature for real-time planning and will be an important next step in the development of this approach. While, SST offers similar theoretical guarantees to the GLC algorithm, the difference in running time is several orders of magnitude in all of the trials making the GLC preferable under most circumstances.

A final remark is that the GLC algorithm suffers from the curse of dimensionality like all other motion planning algorithms. The PSPACE-hardness of these problems suggests this issue will never be resolved. The examples demonstrate the GLC algorithm on state spaces of dimension two to six, and we observe that with increasing dimension, the time required to obtain visually acceptable trajectories increases rapidly with dimension. Example 5.5.4 had six states and required one to five seconds to produce visually acceptable trajectories. This suggests that six states is roughly the limit for real-time applications.
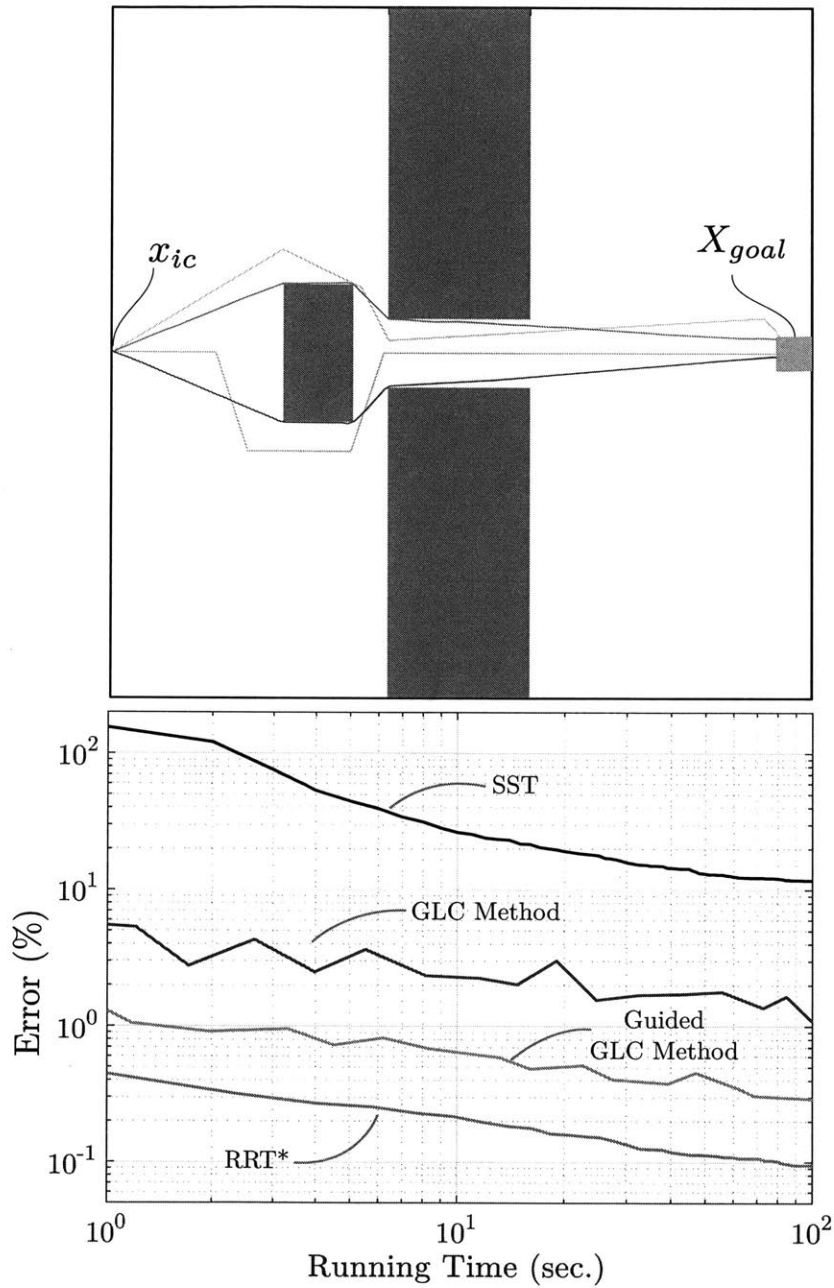
Figure 5-6: (top) The free space X_{free} is shown in white with obstacles shown in grey. The green square is the goal region X_{goal}. The light red and blue paths are low resolution solutions obtained by the RRT* method and the GLC method respectively. The dark colored paths correspond to high resolution solutions. (bottom) The running time required by each algorithm to produce a solution of a particular accuracy is shown. The SST method is shown in black, the GLC method in blue, and the RRT* method in red. Additionally, a guided implementation of the GLC method utilizing the RRT* steering solution as a heuristic is shown in green.
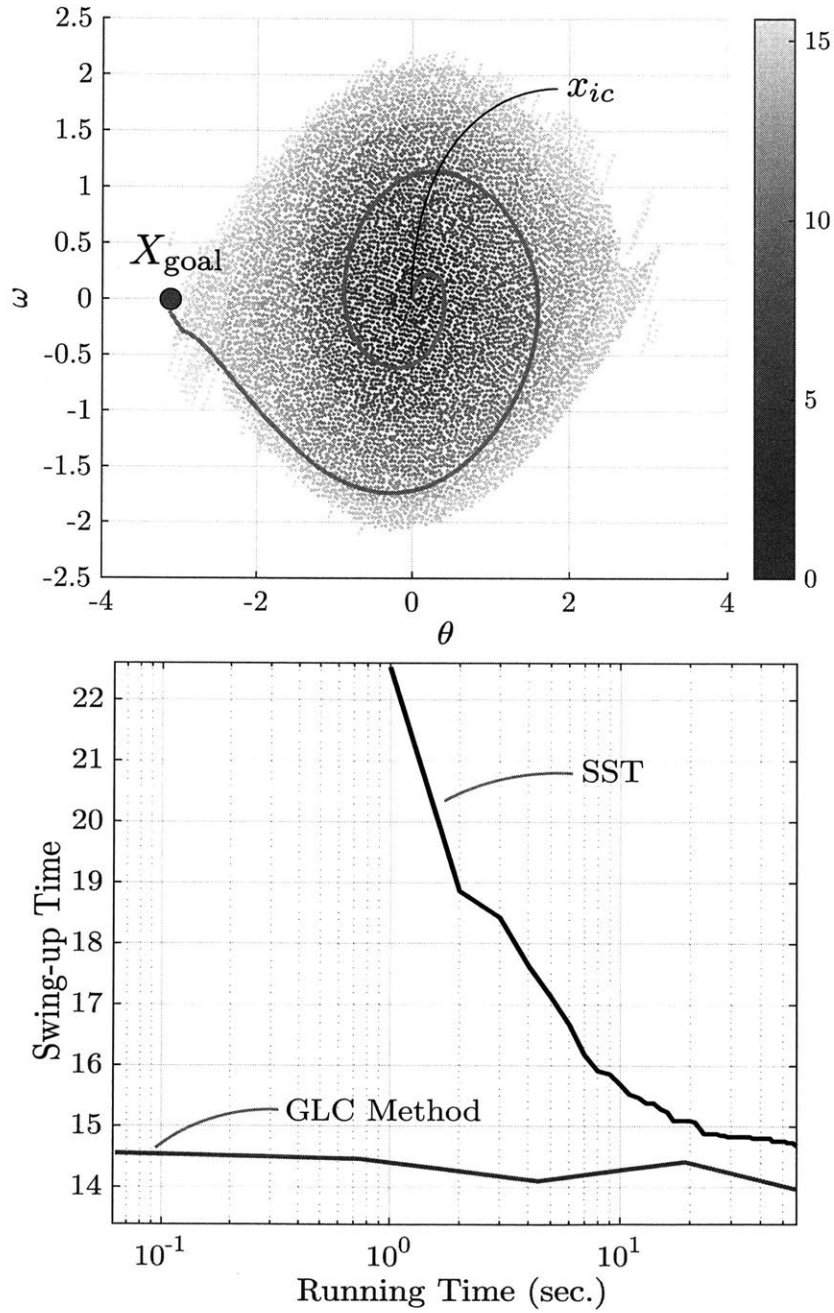
Figure 5-7: (top) The state space trajectory of the pendulum swing-up motion is shown in red. Colored markers show terminal states of minimal trajectories which have been evaluated with the color indicating the cost of the trajectory terminating at that point. (bottom) The running-time and cost of of the output to the GLC algorithm for increasing resolution is shown in blue. The 10 trial average of the SST algorithms output from 1 second of run-time to 500 seconds of run-time is shown in black. Note that the swing-up time is dimensionless.

60

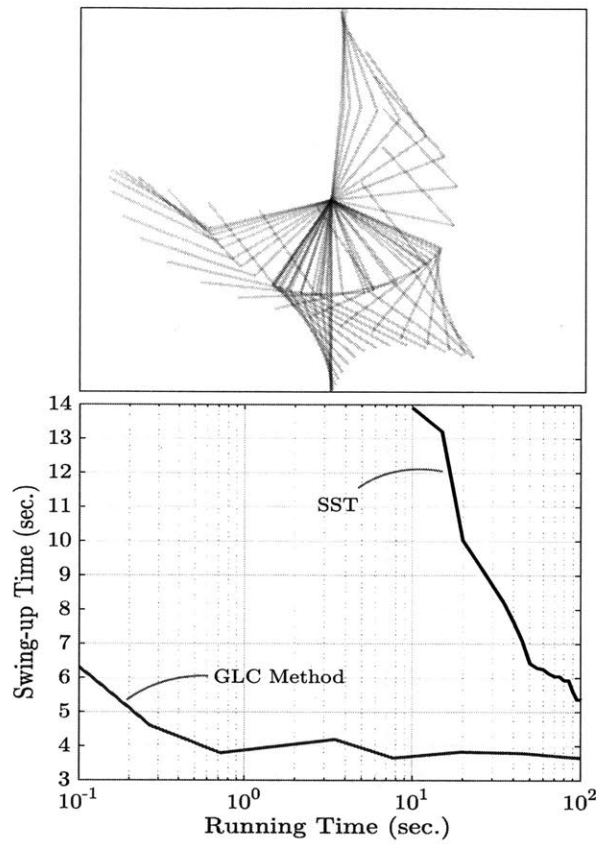Figure 5-8: (top) The swing-up motion of the acrobot returned by the GLC algorithm. (bottom) The running-time and cost of of the output to the GLC algorithm for increasing resolution is shown in blue. The 10 trial average of the SST algorithms output from 1 second of run-time to 100 seconds of run-time is shown in black.

$R = 8$
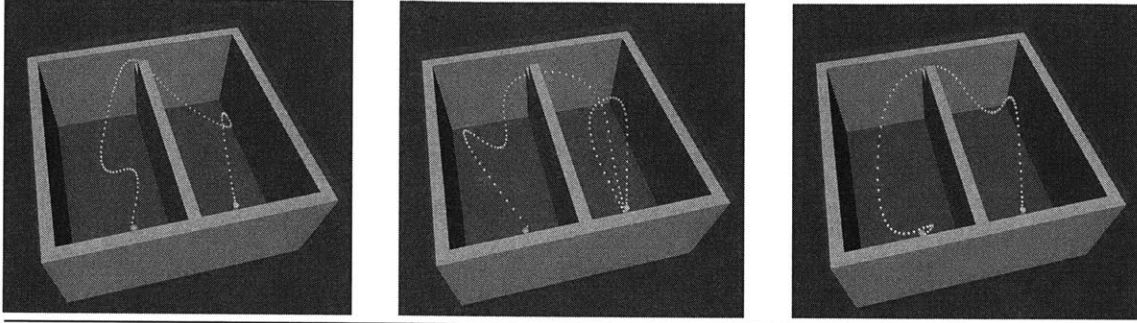
$R = 10$
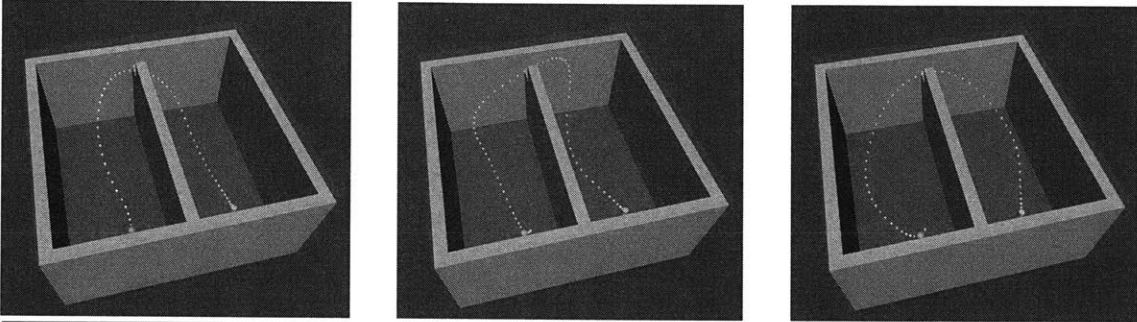
$R = 12$
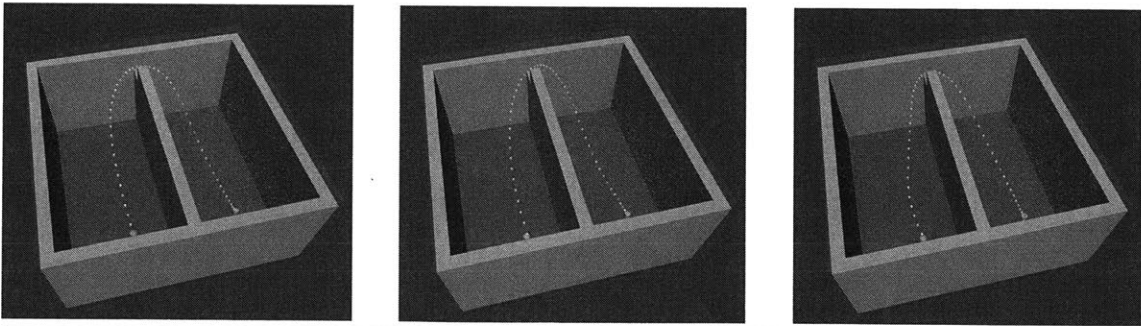
Figure 5-9: The top row of graphics show trajectories beginning at the start configuration (light blue marker) and terminating at the goal configuration (dark blue marker) returned by the GLC algorithm with the resolution $R = 8$. Since the input space Ω_R is deterministic up to a random orthogonal transformation in this example, there is some variation in the output. The middle and bottom graphics show the same results for $R = 10$ and $R = 12$ respectively. With increasing resolution the trajectories converge to the optimal trajectory.
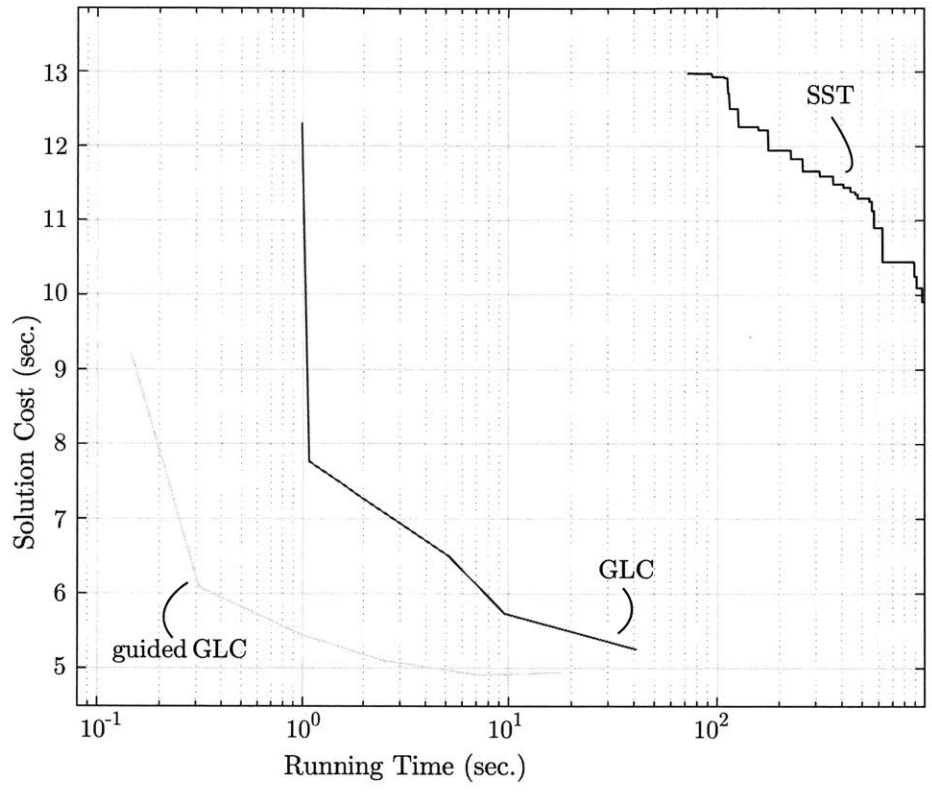
Figure 5-10: The running time of the GLC algorithm (blue) and a guided variant (green) for $R \in \{8, 9, 10, 11, 12\}$ are compared to the SST algorithm (black).
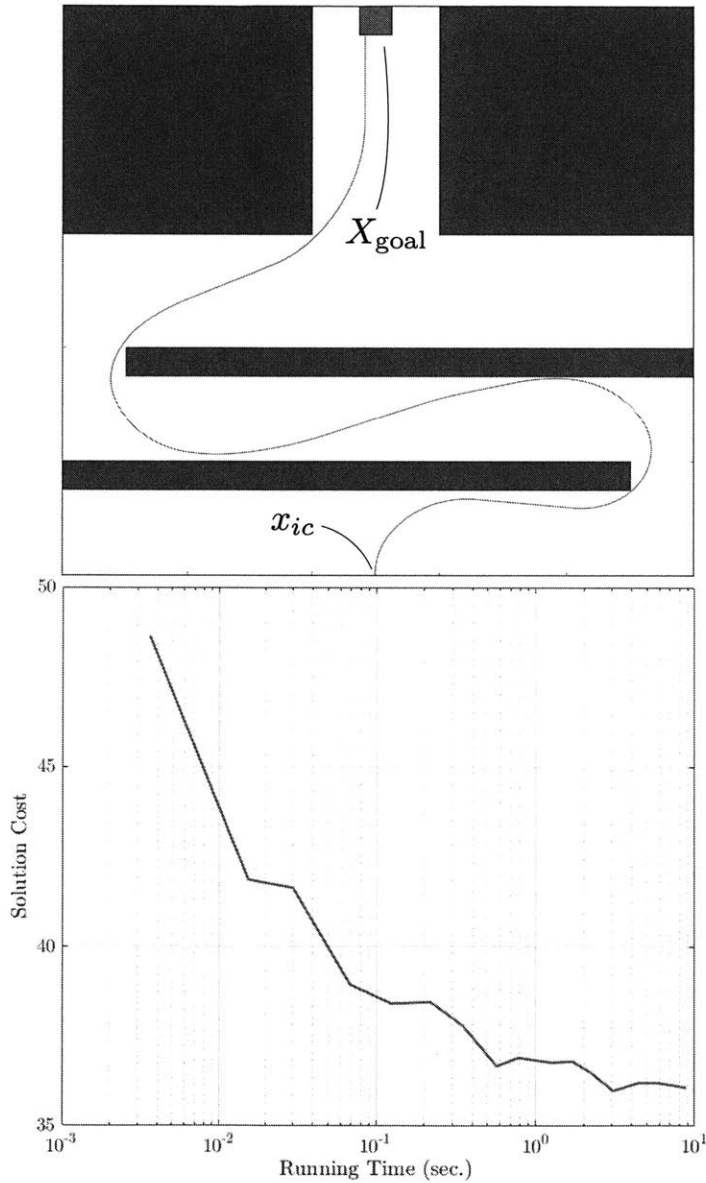
Figure 5-11: (top) The blue curve illustrates a trajectory from the initial configuration to the goal set which minimizes a mixed penalty on time and lateral acceleration which reflects a comfort objective for a passenger vehicle. (bottom) The running time of the GLC algorithm required to obtain a solution of given cost.

Chapter References

[Bel56] Richard Bellman, *Dynamic programming and Lagrange multipliers*, Proceedings of the National Academy of Sciences **42** (1956), no. 10, 767–769.

[Ber95] Dimitri P Bertsekas, *Dynamic programming and optimal control*, vol. 1, Athena Scientific Belmont, MA, 1995.

[HNR68] Peter E Hart, Nils J Nilsson, and Bertram Raphael, *A formal basis for the heuristic determination of minimum cost paths*, Systems Science and Cybernetics **4** (1968), no. 2, 100–107.

[Kar] Sertac Karaman, *RRT* Library*, `http://karaman.mit.edu/software.html`, Accessed: Jan. 2016.

[LLB] Yanbo Li, Zakary Littlefield, and Kostas E Bekris, *Sparse RRT package*, `https://bitbucket.org/pracsys/sparse_rrt/`, Accessed: Jan. 2016.

[PF16] Brian Paden and Emilio Frazzoli, *A generalized label correcting method for optimal kinodynamic motion planning*, Algorithmic Foundations of Robotics XII (WAFR), 2016, Available at: `http://www.wafr.org/program.html#detailedprogram`.

[PF17] Brian Paden and Emilio Frazzoli, *Selection of input primitives for the generalized label correcting method*, American Control Conference (ACC), 2017.

[Spo95] Mark W Spong, *The swing up control problem for the acrobot*, Control Systems, IEEE **15** (1995), no. 1, 49–55.

Chapter 6

Admissible Heuristics for Optimal Kinodynamic Motion Planning

Many graph search problems arising in robotics and artificial intelligence that would otherwise be intractable can be solved efficiently with an effective heuristic informing the search. However, efficiently obtaining a shortest path on a graph requires the heuristic to be admissible as described in the seminal paper introducing the A* algorithm [HNR68]. In short, an admissible heuristic provides an estimate of the optimal cost to reach the goal from every vertex, but never overestimates the optimal cost. A good heuristic is one which closely underestimates the optimal cost-to-go from every vertex to the goal.

The workhorse heuristic in shortest path problems is the Euclidean distance from a given state to the goal. This heuristic is admissible irrespective of the obstacles in the environment since, in the complete absence of obstacles, the length of the shortest path from a particular state to the goal is the Euclidean distance between the two points. Returning to example 5.5.1 of the last chapter, using the Euclidean distance as a heuristic in the GLC algorithm reduces the number of iterations required by 84% in comparison to a uniform cost search. Figure 6-1 shows a side-by-side comparison of the region of X_{free} explored by the informed search versus the uniform cost search.

The value of admissible heuristics for kinodynamic planning has already been identified in a number of works where admissible heuristics for individual problems have been derived and used to reduce the running time of planning algorithms [GSB14, GSB15, HZ15, PF16]. The GLC method is among these.

While admissibility of a heuristic is an important concept it gives rise to two challenging questions in the context of kinodynamic motion planning. First, without a priori knowledge of the optimal cost-to-go, how do we verify the admissibility of a candidate heuristic; and second, how do we systematically construct good heuristics for kinodynamic motion planning problems? The goal of this chapter is to present a principled study of admissible heuristics for kinodynamic motion planning which answers these questions.

The Value Function: The optimal cost-to-go or *value function* $V : X_{\text{free}} \to \mathbb{R}$ is central to this discussion and describes the greatest lower bound on the cost to reach
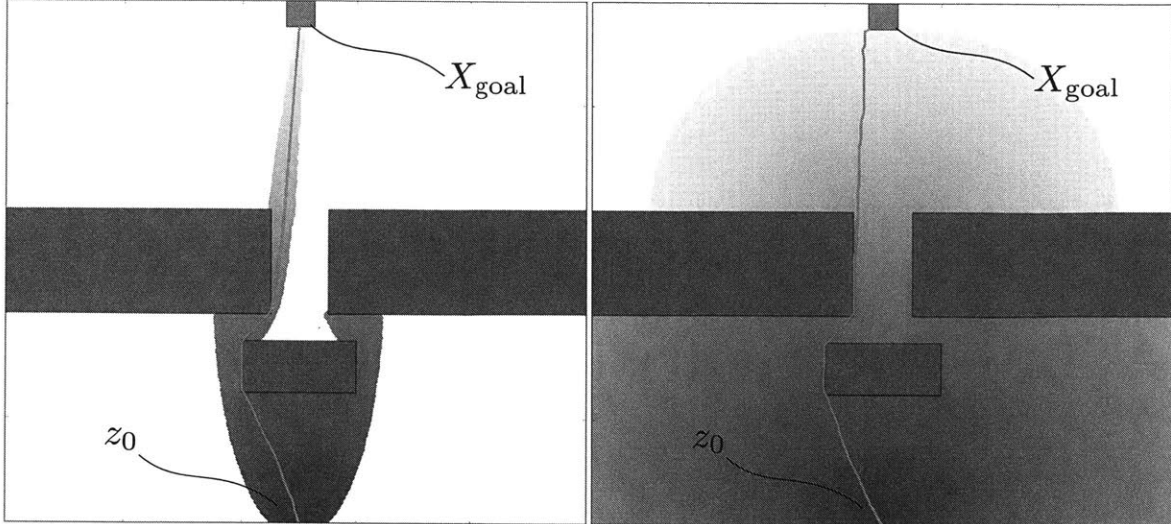
Figure 6-1: Colored markers indicate the terminal state of a trajectory evaluated by the GLC algorithm with color indicating the cost to reach that state. The left tile shows the region explored by the informed GLC algorithm while the right tile is result of the uniform cost search.

the goal set from the initial state $x_0 \in X_{\text{free}}$. That is,

$$V(x_0) = \inf_{u \in \mathcal{U}_{\text{goal}}} \{J_{x_0}(u)\}, \tag{6.0.1}$$

where $\mathcal{U}_{\text{goal}}$ and $\mathcal{X}_{\text{goal}}$ are redefined (with some abuse of notation) with respect to an initial condition x_0. It is important to note that this definition of V implies that V is well defined and unique. The following properties of V follow immediately from the assumption $g(z, w) \geq 0$ in (2.2.4):

$$V(z) \geq 0 \quad \forall z \in X_{\text{free}}, \qquad V(z) = 0 \quad \forall z \in X_{\text{goal}}. \tag{6.0.2}$$

It is well known in optimal control theory that when the value function is differentiable[1], obtaining $V(x_0)$ in (6.0.1) for every x_0 in X_{free} is equivalent to solving the celebrated *Hamilton-Jacobi-Bellman equation*,

$$\inf_{w \in \Omega} \{\langle \nabla V(z), f(z, w)\rangle + g(z, w)\} = 0, \quad \forall z \in X_{\text{free}} \setminus \bar{X}_{\text{goal}}, \tag{HJB}$$

with the boundary condition $V(z) = 0$ for all z in the closure of X_{goal}. In general solving this equation is equivalent to solving Problem 2.2.1 from every initial condition in X_{free} which is considerably more difficult.

---

[1]The equivalence holds even when the value function is not differentiable if a generalized solution concept known as a *viscosity solution* is used [CL83].

6.1 Graph-Based Approximations

Most kinodynamic motion planning algorithms, the GLC algorithm included, structure an approximation of the problem as a graph $(\mathcal{V}, E)$ with paths in the graph corresponding to trajectories satisfying the dynamic model (2.1.3). Conceptually, shortest paths on the graph are in some sense faithful approximations of optimal feasible trajectories for the problem. This was analyzed for the GLC algorithm in Section 4.1.

The non-negativity of the cost function (2.2.4) enables a nonnegative edge-weight to be assigned to each edge corresponding to the cost of the trajectory in relation with that edge. The approximated problem can then be addressed using shortest path algorithms for graphs.

The value function $\hat{V} : \mathcal{V} \to \mathbb{R}$ on the weighted graph is analogous to the value function V in the original problem. For a vertex z in the graph, $\hat{V}(z)$ is the cost of a shortest path to one of the goal vertices: $\mathcal{V} \cap X_{\text{goal}}$. Since the feasible trajectories represented by the graph are a subset of the feasible trajectories of the problem we have the inequality

$$V(z) \leq \hat{V}(z) \qquad \forall z \in \mathcal{V}. \tag{6.1.1}$$

6.1.1 Admissible Heuristics

Recall that the GLC algorithm utilized a priority queue which evaluated candidate paths in a graph in order of their cost in (5.3.1). Evaluating candidates in some order of merit is known as a best first search. This is accomplished with an operation which is traditionally called "pop" in software libraries[2]. When we have a *heuristic* $H : X_{\text{free}} \to \mathbb{R}$ that estimates the remaining cost to reach the goal, the ordering of the priority queue is modified so that the pop operation returns an element of the queue Q satisfying,

$$\text{pop}(Q) \in \underset{u \in Q}{\text{argmin}}\{J_{x_{ic}}(u) + H([\varphi_{x_{ic}}(u)](\tau(u)))\}. \tag{6.1.2}$$

This orders the priority queue according to the cost of the input signal u plus the estimate of the remaining cost to reach the goal from the terminal state of the trajectory associated to the input signal $[\varphi_{x_{ic}}(u)](\tau(u))$.

A heuristic H for a problem with value function V is *admissible* if it satisfies

$$H(z) \leq V(z) \qquad \forall z \in X_{\text{free}}. \tag{6.1.3}$$

When a heuristic is admissible, the first signal found to be in $\mathcal{U}_{\text{goal}}$ is an optimal signal within the approximation $\mathcal{U}_R$. In light of (6.1.1), an admissible heuristic for the kinodynamic motion planning problem will also be admissible for any graph-based approximation to the problem. For the remainder of this chapter, the set of candidate heuristics will be restricted to differentiable scalar valued functions on X_{free}.

---

[2]This operation is technically not a function in the usual sense. Rather, it is a relation.

Another important concept related to heuristics is referred to as consistency[3]. Consistency of a heuristic is similar to the triangle inequality for metric spaces (cf. Appendix B.1). To define consistency, the value function for the kinodynamic motion planning problem must be parametrized by the goal set. This is denoted $V(z; X_{\text{goal}})$. A heuristic H is *consistent* if,

$$H(z) = 0, \quad \forall z \in X_{\text{goal}},$$

$$H(z) \leq V(z; \{y\}) + H(y), \quad \forall y, z \in X_{\text{free}}. \tag{6.1.4}$$

Note that the inequality above involves the optimal cost-to-go from z to y. While a best first search with an admissible heuristic will return an optimal solution, if the heuristic is not consistent, then the search may not benefit from using the heuristic in terms of iterations required to terminate the search.

Since the value function is unknown it is difficult to check that (6.1.3) and (6.1.4) are satisfied for a particular heuristic H. The next result characterizes a convex subset of admissible heuristics. This sufficient condition for admissibility can be checked directly from the problem data.

Theorem 6.1.1. *A heuristic H is admissible if the following two conditions are satisfied:*

$$H(z) \leq 0 \quad \forall z \in X_{\text{goal}}, \tag{AH1}$$

$$\langle \nabla H(z), f(z, w) \rangle + g(z, w) \geq 0, \quad \forall z \in X_{\text{free}} \setminus cl(X_{\text{goal}}), \text{ and } \forall w \in \Omega. \tag{AH2}$$

Further, the heuristic is consistent if

$$H(z) = 0 \quad \forall z \in X_{\text{goal}}, \tag{CH1}$$

Proof. (admissibility) Let u be an input signal resulting in a feasible trajectory x terminating in the goal. By construction the terminal state $x(\tau(u))$ is in the goal set X_{goal}. Therefore, $H(x(\tau(u))) \leq 0$ by (AH1). This implies the inequality

$$H(x(0)) \leq H(x(0)) - H(x(T)). \tag{6.1.5}$$

Then by the fundamental theorem of calculus

$$H(x(0)) \leq -\int_0^{\tau(u)} \frac{d}{dt} H(x(t)) \, \mu(dt). \tag{6.1.6}$$

Applying the chain rule to the integrand yields

$$H(x(0)) \leq -\int_0^{\tau(u)} \langle \nabla H(x(t)), f(x(t), u(t)) \rangle \, \mu(dt). \tag{6.1.7}$$

---

[3]This usage of the term consistency is distinct from the usage in chapter 4. This is an unfortunate double usage of the term in robotics.

Since x is feasible, $x(t) \in X_{\text{free}}$ for all $t \in [0, \tau(u)]$. Directly applying (AH2),

$$H(x(0)) \leq \int_0^{\tau(u)} g(x(t), u(t)), \mu(dt). \tag{6.1.8}$$

The right hand side is now simply the cost $J_{x_{ic}}(u)$ associated to the input signal u.

$$H(x(0)) \leq J_{x_{ic}}(u). \tag{6.1.9}$$

Since the choice of u was arbitrary within $\mathcal{U}_{\text{goal}}$, we conclude that $H(x(0)) \leq J_{x_{ic}}(u)$ for every $u \in \mathcal{U}_{\text{goal}}$ with associated trajectory x. Equivalently,

$$H(z) \leq V(z) \qquad \forall z \in X_{\text{free}}. \tag{6.1.10}$$

(consistency) Let u be an input signal resulting in a trajectory x from $x(0) = z$ and terminating at $x(\tau(u)) = y$. By a similar line of reasoning as for the admissibility argument observe that

$$
\begin{aligned}
H(x(0)) - H(x(\tau(u))) &= -\int_0^{\tau(u)} \frac{d}{dt} H(x(t)) \, \mu(dt) \\
&= -\int_0^{\tau(u)} \langle \nabla H(x(t)), f(x(t), u(t)) \rangle \, \mu(dt) \\
&\leq \int_0^{\tau(u)} g(x(t), u(t) \, \mu(dt) \\
&= J(x, u).
\end{aligned}
\tag{6.1.11}
$$

Thus, $H(z) - H(y)$ lower bounds $J_{x_0}(u)$ for any input signal resulting in a trajectory from z to y. Since $V(\cdot\,; \{y\})$ is the greatest lower bound to the cost of such trajectories we have

$$H(z) - H(y) \leq V(z; \{y\}), \qquad \forall y, z \in X_{\text{free}}. \tag{6.1.12}$$

Rearranging the expression above yields the definition of consistency for $H(\cdot\,; X_{\text{goal}})$. $\qquad \square$

It is interesting to note that the admissibility (and consistency) conditions are affine constraints which implies that the subset of admissible heuristics satisfying (AH1) and (AH2) are convex.

Corollary 6.1.1. *The set of candidate heuristics satisfying the admissibility conditions* (AH1) *and* (AH2) *is convex.*

Proof. Choose $\lambda \in [0, 1]$, and heuristics H_1 and H_2 satisfying (AH1) and (AH2). The convex combination is given by $\lambda H_1 + (1 - \lambda)H_2$. By (AH1), both $H_1(z)$ and $H_2(z)$ are less than or equal to zero for all z in X_{goal}. Thus,

$$\lambda H_1(z) + (1 - \lambda)H_2(z) \leq 0 \qquad \forall z \in X_{\text{goal}}. \tag{6.1.13}$$

Then inserting the convex combination into (AH2) yields

$$
\begin{aligned}
&\langle \nabla(\lambda H_1(z) + (1-\lambda) H_2(z)), f(z,w)\rangle + g(z,w) \\
&= \lambda \langle \nabla H_1(z), f(z,w)\rangle + (1-\lambda) \langle \nabla H_2(z), f(z,w)\rangle + g(z,w) \\
&= \lambda \left(\langle \nabla H_1(z), f(z,w)\rangle + g(z,w)\right) + (1-\lambda)\left(\langle \nabla H_2(z), f(z,w)\rangle + g(z,w)\right) \\
&\geq 0
\end{aligned}
\tag{6.1.14}
$$

Since H_1 and H_2 individually satisfy (AH2), $\lambda \geq 0$, and $1 - \lambda \geq 0$, the convex combination of H_1 and H_2 also satisfies (AH2). $\qquad\square$

It is interesting that the set of admissible heuristics is convex, but the fact that this characterization is only a subset of all possible heuristics raises the question of whether there are any good heuristics (close to the value function V) contained in this subset. The next section discusses the usefulness of this set showing that it contains the value function.

6.2   Selecting an Admissible Heuristic

Considering that a good heuristic is one which closely underestimates the value function and that the admissibility conditions of Theorem 6.1.1 characterize a set of such underestimates, a natural approach to the selection of a heuristic would be to find an admissible heuristic whose integral on X_{free} is maximized[4]. Since the constraints are affine and this would be a linear functional of H, the heuristic selection is then a (infinite-dimensional) linear program:

$$
\boxed{
\begin{aligned}
&\max_{H} \int_{X_{\text{free}}} H(z)\, m(dz) \\
&\text{subject to} : H(z) \leq 0 \qquad \forall z \in X_{\text{goal}}, \\
&\qquad\qquad \langle \nabla H(z), f(z,w)\rangle + g(z,w) \geq 0 \\
&\qquad\qquad \forall z \in X_{\text{free}} \setminus cl(X_{\text{goal}}), \text{ and } \forall w \in \Omega.
\end{aligned}
}
\tag{LP}
$$

To justify this formulation, the next result shows that it is equivalent to the HJB equation and the solution to this linear program is in fact the value function.

Theorem 6.2.1. *A function V is the solution to the linear optimization LP if and only if it is the solution to the Hamilton-Jacobi-Bellman equation HJB.*

Proof. ($\Leftarrow$) Let V be a solution to (HJB). Then V is equal to the value function so every admissible heuristic H must satisfy $H(z) \leq V(z)$ on X_{free}. Therefore, the

---

[4]Note that the measure m must be finite on X_{free}.

solution to (HJB) upper bounds the objective to (LP),

$$\int_{X_{\text{free}}} H(z)\, m(dz) \le \int_{X_{\text{free}}} V(z)\, m(dz). \tag{6.2.1}$$

Next, observe that a solution to (HJB) satisfies the admissibility conditions (AH1) and (AH2): The boundary condition $V(z) = 0$ on $cl(X_{\text{goal}})$ implies V satisfies (AH2) and the equation

$$\inf_{w \in \Omega} \left\{ \langle \nabla V(z), f(z, w) \rangle + g(z, w) \right\} = 0, \quad \forall z \in X_{\text{free}} \setminus cl(X_{\text{goal}}), \tag{6.2.2}$$

implies

$$\langle \nabla V(z), f(z, w) \rangle + g(z, w) \ge 0 \quad \forall z \in X_{\text{free}} \setminus cl(X_{\text{goal}}), \text{ and } \forall w \in \Omega, \tag{6.2.3}$$

which is (AH2). Thus, the upper bound in (6.2.1) is attained at the solution to (HJB).

($\Rightarrow$) As a point of contradiction, suppose that H is a solution to (LP) and is not a solution to (HJB). Then there exists $\tilde{z}$ such that

$$\langle \nabla H(\tilde{z}), f(\tilde{z}, w) \rangle + g(\tilde{z}, w) > 0 \quad \forall w \in \Omega. \tag{6.2.4}$$

The existence of a point where there is strict inequality comes from the assumption that H does not solve (HJB), but satisfies the constraints of (LP). By the continuity of H, there is a neighborhood $\mathcal{N}_{\tilde{z}}$ of $\tilde{z}$ on which (6.2.4) holds. Let K be a compact subset of $\mathcal{N}_{\tilde{z}}$ and denote the minimum of H attained on this compact set by ε. Additonally, denote the supremum of $\|f(z, w)\|_2$ over $K \times \Omega$ by F. Then

$$\langle \nabla H(z), f(z, w) \rangle + g(z, w) - \varepsilon \ge 0 \quad \forall z \in K, \text{ and } \forall w \in \Omega. \tag{6.2.5}$$

Next, let $\Psi : X_{\text{free}} \to \mathbb{R}^n$ be a smooth, nonnegative bump function supported on K satisfying $\Psi(x) \le \varepsilon$, and $\|\nabla \Psi(x)\|_2 \le \min\{1, \varepsilon/F\}$. Perturbing H by Ψ yields

$$
\begin{aligned}
&\langle \nabla(H(z) + \Psi(z)), f(z, w) \rangle + g(z, w) \\
&= \langle \nabla H(z), f(z, w) \rangle + \langle \nabla \Psi(z), f(z, w) \rangle + g(z, w) \\
&\ge \langle \nabla H(z), f(z, w) \rangle - \|\nabla \Psi(z)\|_2 \|f(z, w)\|_2 + g(z, w) \\
&\ge \langle \nabla H(z), f(z, w) \rangle - \min\{1, \varepsilon/F\} \cdot F + g(z, w) \\
&\ge \langle \nabla H(z), f(z, w) \rangle - \varepsilon + g(z, w), \quad \forall z \in K, \text{ and } \forall w \in \Omega.
\end{aligned}
\tag{6.2.6}
$$

Thus, the perturbation by Ψ leaves the heuristic admissible by (6.2.5) and this perturbation is a strict improvement to the objective in (LP) a contradiction since H solves (LP). $\qquad \square$

Thus, (HJB) and (LP) can both be solved to obtain the exact optimal cost-to-go for a problem. This equivalence between (HJB) and (LP) is a stronger version of the

result than can be found in the published work [PVF17]. While the HJB equation is a nonlinear partial differential equation, the problem (LP) is a linear program lending itself to the methods of convex analysis.

6.2.1 Problem Relaxations

In many applications the set X_{free} is not entirely known a priori. This is particularly true when the heuristic is computed off-line and used in a real-time application where a perception system constructs or modifies X_{free} for the current task (e.g. detecting obstacles in a robot workspace).

This consideration motivates the following observation: suppose X_{free}, X_{goal}, Ω, f, and g is problem data for problem P; and $\tilde{X}_{\text{free}}$, $\tilde{X}_{\text{goal}}$, $\tilde{\Omega}$, $\tilde{g}$, and $\tilde{f}$ is problem data for problem $\tilde{P}$. Let V_P and $V_{\tilde{P}}$ denote the optimal cost-to-go for each of these problems.

If the two problems are related by

$$X_{\text{free}} \subset \tilde{X}_{\text{free}}, \quad X_{\text{goal}} \subset \tilde{X}_{\text{goal}}, \quad \Omega \subset \tilde{\Omega}, \quad f = \tilde{f},$$
$$g(z, w) \geq \tilde{g}(z, w) \quad \forall z \in X_{\text{free}}, \text{ and } w \in \Omega, \tag{6.2.7}$$

then any feasible trajectory of problem P must also be feasible for problem $\tilde{P}$. Additionally, this trajectory will have the same or lesser cost for $\tilde{P}$. Therefore, $V_{\tilde{P}}(z) \leq V_P(z)$ for any $z \in \mathcal{X}_{\text{free}}$. We can conclude that an admissible heuristic $H_{\tilde{P}}$ for problem $\tilde{P}$ must also be admissible for problem P since $H_{\tilde{P}}(z) \leq V_{\tilde{P}}(z)$ implies $H_{\tilde{P}}(z) \leq V_P(z)$. Problem $\tilde{P}$ is referred to as a *relaxation* of problem P.

In the case of an unknown environment, one can derive an admissible heuristic for a relaxed problem which considers only constraints known in advance. This heuristic remains admissible if X_{free} is updated to a smaller set due to perceived obstacles. Alternatively, it may be easier to verify the admissibility of a candidate heuristic with (AH1) and (AH2), or evaluate (LP) on a relaxation of a particular problem known a-priori. This comes at the expense of increasing the gap between the heuristic and the optimal cost-to-go for the actual problem which can make the heuristic less effective for a search-based algorithm but allows for off-line construction.

6.3 Sum-of-Squares (SOS) Approximation to Equation (LP)

One way to tackle (LP) in the case of problem data consisting of semi-algebraic sets and polynomials is a SOS programming approximation.

SOS programming [Par00] is a method of optimizing a functional of a polynomial subject to semi-algebraic constraints. The technique involves replacing semi-algebraic constraints with sum-of-squares constraints which can then be solved as a semi-definite program (SDP). The advantage of this particular approach to approximating (LP) is that the result of the SOS program is guaranteed to satisfy the

admissibility constraint.

6.3.1 Sum-of-Squares Polynomials

A polynomial p in the ring $\mathbb{R}[z]$ in n variables is said to be a sum-of-squares if it can be written as

$$p(z) = \sum_{k=1}^{d} q_k(z)^2, \tag{6.3.1}$$

for polynomials $q_k(z)$. Clearly, $p(z) \geq 0$ for all $z \in \mathbb{R}^n$. Note also that $p(z)$ is a sum-of-squares if and only if it can be written as

$$p(z) = \mathfrak{m}(z)^T Q \mathfrak{m}(z), \tag{6.3.2}$$

for a positive semi-definite matrix Q and the vector of monomials $\mathfrak{m}(z)$ up to degree d. For a polynomial p admitting a decomposition of the form (6.3.2) we write $p \in SOS$.

Equation (6.3.2) is a collection of linear equality constraints between the entries of Q and the coefficients of $p(z)$. Finding entries of a positive semi-definite Q such that the equality constraints are satisfied is then a semi-definite program (SDP). The complexity of finding a solution to this problem using interior-point methods is polynomial in the size of Q.

This method of analyzing polynomial inequalities has had a profound impact in many fields. As a result there are a number of efficient solvers [Stu99, TTT99] and modeling tools [PPP02, Lof04] available.

6.3.2 Optimizing the Heuristic

To proceed with computing a heuristic using the SOS programming framework, the problem data must consist of polynomials and intersections of semi-algebraic sets. Let

$$\begin{aligned} X_{\text{free}} &= \left\{ z \in \mathbb{R}^n : h_z(z) \geq 0 \right\}, \\ \Omega &= \left\{ w \in \mathbb{R}^m : h_w(w) \geq 0 \right\}, \end{aligned} \tag{6.3.3}$$

for vectors of polynomials h_z and h_w with "$\geq$" denoting element-wise inequalities. Assume also that f, g and the candidate heuristic H are polynomials in the state and control variables. Then the admissibility condition (AH1) is a polynomial inequality. To restrict the non-negativity constraint of the heuristic to X_{free} and Ω, we add auxiliary vectors of SOS polynomials $\lambda_z(z) \geq 0$ and $\lambda_w(w) \geq 0$ to the equation as

$$\begin{aligned} \langle \nabla H(z), f(z,w) \rangle + g(z,w) - \langle \lambda_z(z), h_z(z) \rangle - \langle \lambda_w(w), h_w(w) \rangle \geq 0, \\ \forall w \in \mathbb{R}^m, \text{ and } z \in \mathbb{R}^n, \end{aligned} \tag{6.3.4}$$

which trivially implies the positivity of (AH1) over X_{free} and Ω. When H is a polynomial, the objective in (LP) is linear in the coefficients of H. Thus, it is an appropriate objective for an SOS program.

The SOS program which is solved to obtain an admissible heuristic is then

$$\max_{H,\lambda_z,\lambda_w} \int_{X_{\text{free}}} H(z)\,m(dz)$$

$$\text{subject to}: H(z) \le 0, \quad \forall z \in cl(X_{\text{goal}}),$$

$$\langle \nabla H(z), f(z,w) \rangle + g(z,w) - \langle \lambda_z(z), h_z(z) \rangle - \langle \lambda_w(w), h_w(w) \rangle \in SOS,$$

$$\lambda_z(z), \lambda_w(w) \in SOS.$$

$$(6.3.5)$$

6.4 Examples of Heuristics for Kinodynamic Motion Planning

The examples of this section demonstrate how to directly apply Theorem 6.1.1 to verify admissibility of candidate heuristics as well as the heuristic synthesis optimization (LP).

6.4.1 Admissibility of Euclidean Distance in Shortest Path Problems

In the first example we show how to use Theorem 6.1.1 to verify the classic Euclidean distance heuristic for kinematic shortest path problems.

Consider a holonomic shortest path problem,

$$\dot{x}(t) = u(t), \tag{6.4.1}$$

where $X_{\text{free}} = \mathbb{R}^n$, $X_{\text{goal}} = \{0\}$, and $u(t) \in \{w \in \mathbb{R}^n : \|w\| = 1\}$. The running cost is given by

$$g(z,w) = 1. \tag{6.4.2}$$

We would like to verify the heuristic

$$H(z) = \|z\|. \tag{6.4.3}$$

Applying the admissibility Lemma we obtain

$$\langle \nabla H(z), f(z,w) \rangle + g(z,w) = \frac{\langle z, w \rangle}{\|z\|} + 1$$
$$\ge \frac{-\|z\|\|w\|}{\|z\|} + 1 \tag{6.4.4}$$
$$\ge -1 + 1$$
$$= 0,$$

which reverifies the fact that the Euclidean distance is an admissible heuristic for the
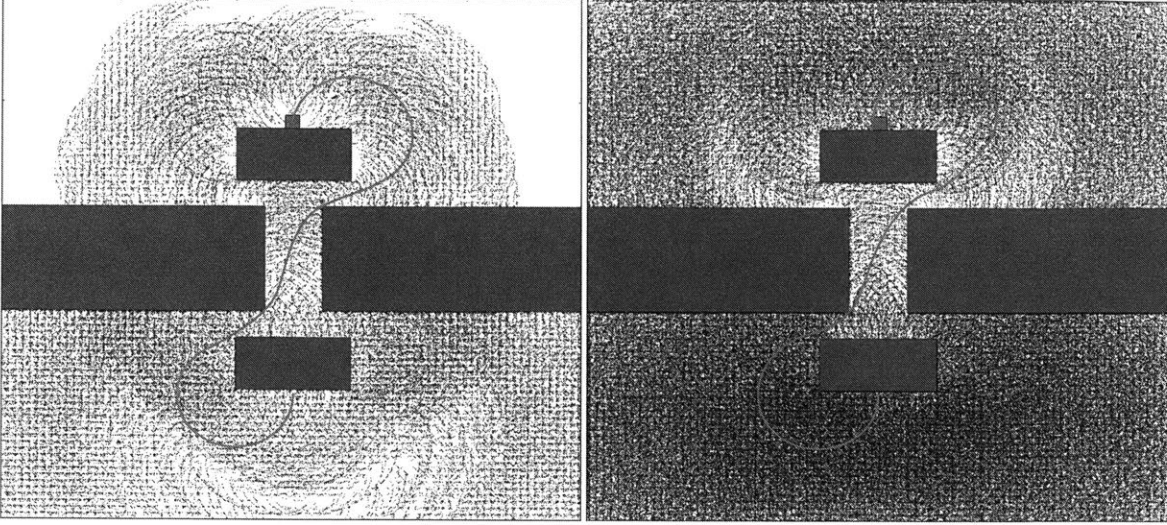
76

Figure 6-2: Comparison of the vertices examined by the heuristically guided search (left) and the uniform cost search (right) for a simple wheeled robot motion planning problem. Equation 6.4.10) was used as the heuristic.

shortest path problem. The crux of this derivation is simply applying the Cauchy-Schwarz inequality. Figure 6-1 illustrates the use of this heuristic in an environment with obstacles.

6.4.2 Wheeled Robot Example

Consider a simple wheeled robot with state coordinates in $\mathbb{R}^3$, and whose mobility is described by

$$\dot{p}_1(t) = \cos(\theta(t)), \quad \dot{p}_2(t) = \sin(\theta(t)), \quad \dot{\theta}(t) = u(t), \qquad (6.4.5)$$

where $(p_1(t), p_2(t), \theta(t))$ are individual coordinates of the state. Let $X_{\text{free}} = \mathbb{R}^3$, $X_{\text{goal}} = \{(0,0,0)\}$, and $\Omega = [-1, 1]$. The running cost

$$g(p_1(t), p_2(t), \theta(t)) = 1, \qquad (6.4.6)$$

results in a cost equal to the path length in the x-y plane.

We will consider two candidate heuristics: the line segment connecting the p_1-p_2 coordinate to the origin (discussed in [DTMD10]), and the magnitude of the heading error,

$$H_1(p_1(t), p_2(t), \theta(t)) = \sqrt{p_1(t)^2 + p_2(t)^2}, \quad H_2(p_1(t), p_2(t), \theta(t)) = |\theta(t)|. \quad (6.4.7)$$

Clearly, (AH1) is satisfied for both heuristics. Then (AH2) is once again verified with the Cauchy-Schwarz inequality.

For brevity, but with some abuse of notation, the time argument is dropped from the trajectory in the following derivations. Inserting the expression for the heuristics

77

into (AH2) yields

$$\langle \nabla H_1(p_1, p_2, \theta), f(p_1, p_2, \theta, u)\rangle + g(p_1, p_2, \theta, u) = \frac{\langle (p_1, p_2, 0), (\cos(\theta), \sin(\theta), u)\rangle}{\sqrt{p_1^2 + p_2^2}} + 1$$

$$\geq -\frac{\sqrt{p_1^2 + p_2^2}\sqrt{\cos^2(\theta) + \sin^2(\theta)}}{\sqrt{p_1^2 + p_2^2}} + 1$$

$$= 0.$$

$$(6.4.8)$$

Similarly, for H_2,

$$\langle \nabla H_2(p_1, p_2, \theta), f(p_1, p_2, \theta, u)\rangle + g(p_1, p_2, \theta, u) = \frac{\langle (0, 0, \theta), (\cos(\theta), \sin(\theta), u)\rangle}{|\theta|} + 1$$

$$\geq -\frac{|\theta||u|}{|\theta|} + 1$$

$$= 0.$$

$$(6.4.9)$$

Therefore, both heuristics are admissible. Note that the maximum of the two heuristics is also an admissible heuristic,

$$H(p_1, p_2, \theta) = \max\{H_1(p_1, p_2, \theta), H_2(p_1, p_2, \theta)\}. \tag{6.4.10}$$

Figure 6-2 illustrates a shortest path query in a simple environment with and without the use of the heuristic in (6.4.10). The heuristically guided search obtains a solution in 81,686 iterations while the uniform cost search requires 403,197 iterations. In contrast, using just H_1 in Equation (6.4.7) as a heuristic requires 104,492 iterations.

6.4.3 Autonomous Underwater Vehicle Example

Consider an autonomous underwater vehicle (AUV) navigating a strong current relative to the vehicle's top speed as discussed in [GBA$^+$14]. This scenario is common for long range underwater gliders which travel at roughly $0.5\,m/s$ relative to currents traveling at 1.0-$1.5\,m/s$.

Let the state space be $\mathbb{R}^n$ ($n = 2$ or 3) representing position in a local Cartesian coordinate system. The AUV's motion is modeled by

$$\dot{x}(t) = c(x(t)) + u(t), \tag{6.4.11}$$

where $c : \mathbb{R}^n \to \mathbb{R}^n$ describes the current velocity at each point in the state space. The input $u(t)$ is the controlled velocity relative to the current. The thrust constraint is represented by

$$\Omega = \{w \in \mathbb{R}^n : \|w\| \leq w_{\max}\}, \tag{6.4.12}$$

where $w_{\max}$ is the maximum achievable speed relative to the current. The goal set is

$X_{\text{goal}} = \{0\}$. The running cost

$$g(x(t), u(t)) = 1 + \|u(t)\|, \tag{6.4.13}$$

reflects a penalty on the duration of the path as well as the total work done by AUV's motors.

The admissibility of the heuristic proposed in [GAO05] can be evaluated using Theorem 6.1.1 . Let $v_{\text{max}} := \max_{z}\{\|c(z)\|\}$ be the maximum speed of the current, and consider the heuristic

$$H(z) = \frac{\|z\|}{(v_{\text{max}} + w_{\text{max}})}. \tag{6.4.14}$$

Clearly (AH1) is satisfied. Now evaluate (AH2),

$$
\begin{aligned}
\langle \nabla H(z), f(z,w) \rangle + g(z,w) &= \frac{\langle z, c(z)+w \rangle}{\|z\|(v_{\text{max}}+w_{\text{max}})} + 1 + \|w\| \\
&\geq \frac{-\|z\| \cdot \|c(z)\| - \|z\| \cdot \|w\|}{\|z\|(v_{\text{max}}+w_{\text{max}})} + 1 + \|w\| \\
&\geq \frac{-\|c(z)\| - \|w\|}{(v_{\text{max}}+w_{\text{max}})} + 1 + \|w\|.
\end{aligned}
\tag{6.4.15}
$$

Notice that v_{max} and w_{max} were defined so that $\|w\| + \|c(z)\| \leq v_{\text{max}} + w_{\text{max}}$. Thus,

$$
\begin{aligned}
\langle \nabla H(z), f(z,w) \rangle + g(z,w) &\geq -1 + 1 + \|w\| \\
&\geq 0,
\end{aligned}
\tag{6.4.16}
$$

so the heuristic is admissible. Figure 6-3 illustrates a numerical example in a 2D environment where the maximum current speed is 2.6 times the AUV's maximum relative speed. The heuristically guided search expands 49% fewer vertices than the uniform cost search.
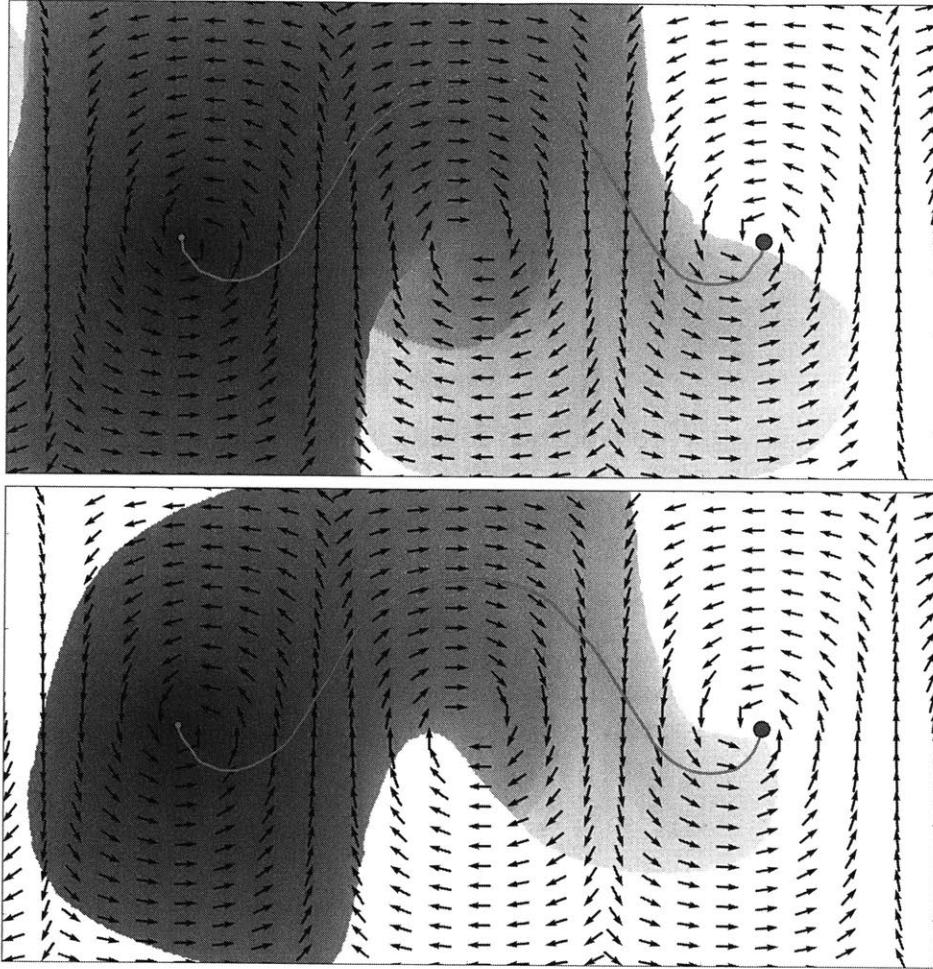
Figure 6-3: Comparison of a minimum cost path search in a graph approximating feasible paths for an AUV in a strong current. The heuristically guided search using equation (6.4.14) of Example 6.4.3 obtains the solution in 271,642 iterations (left) while the uniform cost search requires 537,168 iterations (right). Initial and final states are illustrated with red and blue markers respectively. Colored regions represent states explored by the algorithm and the color indicates the relative cost to reach that state.

6.5 Sum-of-Square Heuristic Synthesis Examples

The last examples demonstrate the SOS programming formulation described in Section 6.3. A closed form solution for the optimal cost-to-go is known for the examples in this section which provides a useful point of comparison for the computed heuristic.

The example problem was implemented using the SOS module in YALMIP [Lof04] and solved using SDPT3 for the underlying semi-definite program [TTT99]. To further illustrate the approach, YALMIP scripts for these examples can be found in [PVF16].

6.5.1 Single Integrator Example

To illustrate the procedure, we revisit Example 6.4.1 in the 1-dimensional case. The differential constraint is given by $f(x, u) = u$ where $x, u \in \mathbb{R}$, $X_{\text{free}} = [-1, 1]$, $\Omega = [-1, 1]$, and $X_{\text{goal}} = \{0\}$. Again we use the minimum time objective where $g(x, u) = 1$. The value function $V(x) = |x|$ is obtained by inspection.

The heuristic is parameterized by the coefficients of a univariate polynomial of degree $2d$

$$H(x) = \sum_{i=0}^{2d} c_i x^i. \tag{6.5.1}$$

Using a discrete measure on $[-1, 1]$ concentrated at the boundary the SOS program is,

$$
\begin{aligned}
\max_{H, \lambda_x, \lambda_u} \ & \{H(1) + H(-1)\} \\
\text{subject to} : \ & H(0) = 0, \\
& \left(\tfrac{d}{dx} H(x)\right) u + 1 - \lambda_x(x)(1 - x^2) - \lambda_u(u)(1 - u^2) \in SOS, \\
& \lambda_x(x), \lambda_u(u) \in SOS.
\end{aligned}
\tag{6.5.2}
$$

The numerical solution for polynomial heuristics with increasing degree is shown in Figure 6-4.

6.5.2 Double Integrator Example

Consider a double integrator model,

$$\dot{x}_1(t) = x_2(t), \quad \dot{x}_2(t) = u(t), \tag{6.5.3}$$

with the minimum time running cost

$$g(x_1(t), x_2(t), u(t)) = 1, \tag{6.5.4}$$

free space $X_{\text{free}} = [-3, 3]^2$, input space $\Omega = [-1, 1]$, and goal set $X_{\text{goal}} = \{0\}$.
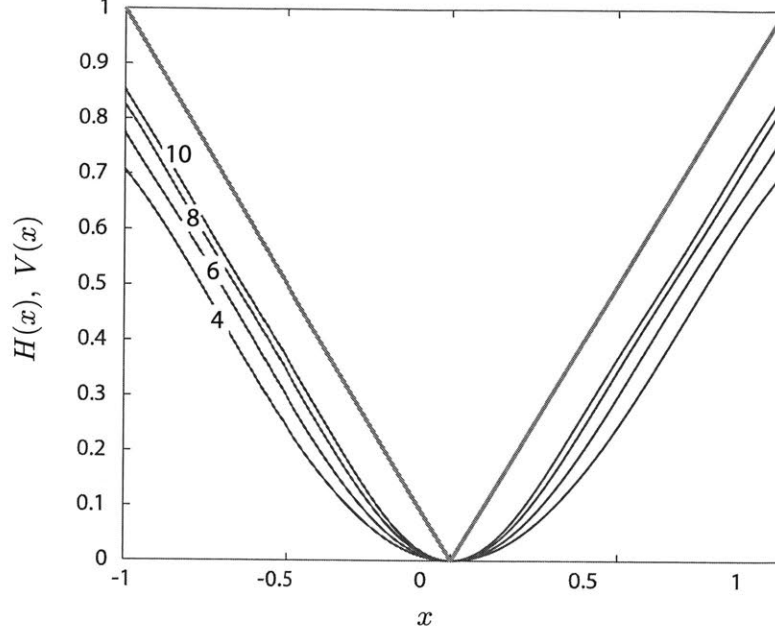
Figure 6-4: Univariate polynomial heuristics of degrees 4, 6, 8, and 10 for the 1D single integrator shown in blue. The value function is shown in red. Polynomial heuristics with higher degree provide better underestimates of the value function.

Polynomial heuristics of degree $2d$ of the form

$$H(x_1, x_2) = \sum_{p+q \leq 2d} c_{p,q} \, x_1^p x_2^q, \tag{6.5.5}$$

are computed with the measure m in (LP) supported on $[-2, 2] \times [-\sqrt{2}, \sqrt{2}]$. This focuses the optimization in a region around the goal while maintaining admissibility of the heuristic over all of X_{free}.

The SOS approximation of (LP) is formulated as follows

$$\max_{H, \lambda_x, \lambda_u} \int_{X_{\text{free}}} H(x_1, x_2) \, m(dx)$$

$$\text{subject to}: H(0, 0) \leq 0,$$

$$\langle \nabla H(x_1, x_2), (x_2, u) \rangle + 1$$

$$-\lambda_{x_1}(x_1) \, (9 - x_1^2)$$

$$-\lambda_{x_2}(x_2) \, (9 - x_2^2) \tag{6.5.6}$$

$$-\lambda_u(u)(1 - u^2) \in SOS,$$

$$\lambda_{x_1}(x_1), \lambda_{x_2}(x_2), \lambda_u(u) \in SOS.$$

Note that the polynomial H can be integrated over the rectangular region S in closed form with standard integration rules or with the integration functionality in YALMIP.

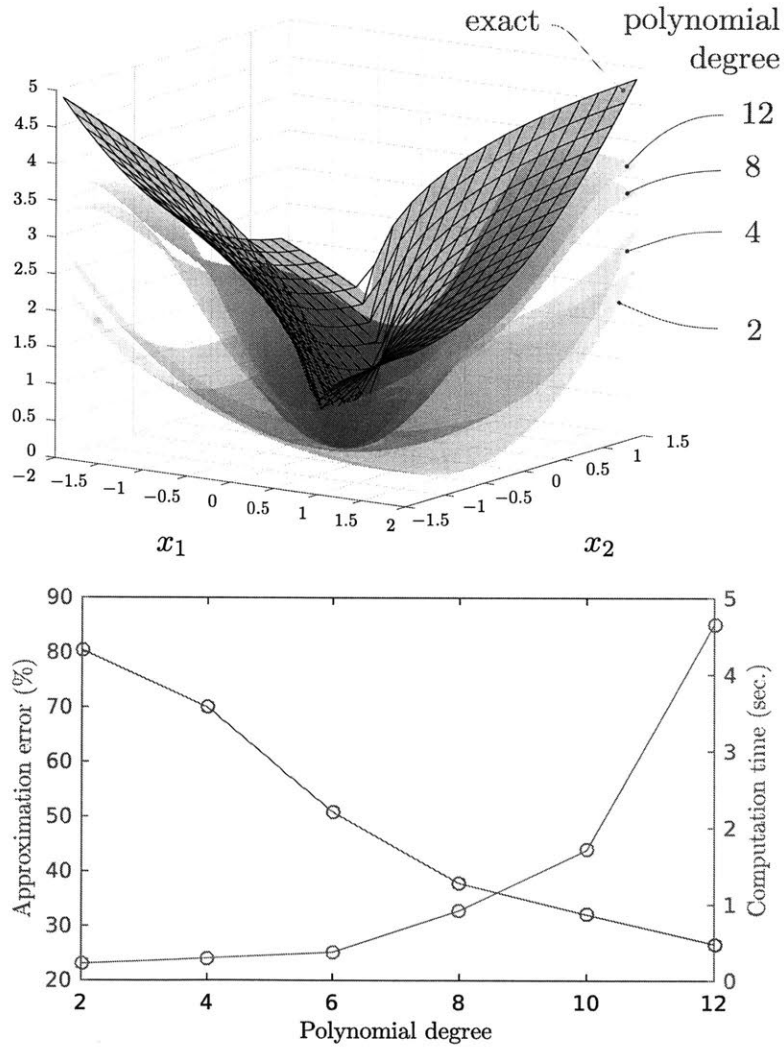The optimized heuristics of increasing degree are shown in Figure 6-5 together

Figure 6-5: (top) Polynomial heuristics of degree 2, 4, 8, and 12 for the double integrator model in comparison with the known value function shown in red. (bottom) Running time and approximation accuracy to compute heuristics of degrees from 2 to 12.

with the value function for $X_{\text{free}} = \mathbb{R}^n$. The percent error in Figure 6-5 is defined as

$$\% \text{ error } = 100 \cdot \int_{X_{\text{free}}} \frac{V(z) - H(z)}{V(z)} m(dz). \qquad (6.5.7)$$

6.5.3 Observations

The optimization of Example 6.5.2 focused on the region $[-2, 2] \times [-\sqrt{2}, \sqrt{2}]$ instead of $[-3, 3]^2$. The reason is that some states in $[-3, 3]^2$ cannot be reached and have infinite cost. A remarkable observation is that the resulting SOS program does not admit a maximum when the integral includes a subset of X_{free} where the value function is unbounded. This is consistent with the theoretical results since the heuristic is free to go unbounded over this set as well. Therefore, it is important to optimize the heuristic over a region where the optimal cost-to-go is known to be bounded.

While semi-definite programs run in polynomial time in problem size, there is combinatorial growth in the number of monomials with increasing degree resulting in large semi-definite programs. For an n-dimensional state space, the number of monomials of degree d or less is

$$\binom{n+d}{n}. \qquad (6.5.8)$$

This rapid growth in problem size is one of the current limitations of this approach to (LP). However, the technique produces a reasonable polynomial approximation that is guaranteed to be admissible.

There are a number possibilities for improving this technique. The DSOS and SDSOS [AM14] programming techniques which are being developed will enable the derivation of heuristics with higher degree polynomials. Additionally, a straight forward solution is to apply finite difference methods to directly approximate (LP) as a finite-dimensional linear program, but the strict admissibility is lost in this case.

Chapter References

[AM14] Amir Ali Ahmadi and Anirudha Majumdar, *DSOS and SDSOS optimization: LP and SOCP-based alternatives to sum-of-squares optimization*, Conference on Information Sciences and Systems, IEEE (2014), 1–5.

[CL83] Michael G Crandall and Pierre-Louis Lions, *Viscosity solutions of Hamilton-Jacobi equations*, Transactions of the American Mathematical Society **277** (1983), no. 1, 1–42.

[DTMD10] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel, *Path planning for autonomous vehicles in unknown semi-structured environments*, The International Journal of Robotics Research (IJRR) **29** (2010), no. 5, 485–501.

[GAO05] Bartolome Garau, Alberto Alvarez, and Gabriel Oliver, *Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an A* approach*, Proceedings of the international conference on robotics and automation (ICRA), IEEE, 2005, pp. 194–198.

[GBA$^+$14] Bartolomé Garau, Matias Bonet, Alberto Alvarez, Simón Ruiz, and Ananda Pascual, *Path planning for autonomous underwater vehicles in realistic oceanic current fields: Application to gliders in the western Mediterranean Sea*, Journal of Maritime Research **6** (2014), no. 2, 5–22.

[GSB14] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, *Informed RRT*: Optimal incremental path planning focused through an admissible ellipsoidal heuristic*, International Conference on Intelligent Robots and Systems, IEEE (2014), 2997–3004.

[GSB15] Jonathan D Gammell, Siddhartha S Srinivasa, and Timothy D Barfoot, *Batch informed trees (BIT*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs*, International Conference on Robotics and Automation, IEEE (2015), 3067–3074.

[HNR68] Peter E Hart, Nils J Nilsson, and Bertram Raphael, *A formal basis for the heuristic determination of minimum cost paths*, Transactions on Systems Science and Cybernetics, IEEE **4** (1968), no. 2, 100–107.

[HZ15] Kris Hauser and Yilun Zhou, *Asymptotically optimal planning by feasible kinodynamic planning in state-cost space*, arXiv e-print (2015), Available at: http://arxiv.org/abs/1505.04098.

[Lof04] Johan Lofberg, *YALMIP: A toolbox for modeling and optimization in MATLAB*, International Symposium on Computer Aided Control Systems Design, IEEE, 2004, pp. 284–289.

[Par00] Pablo A Parrilo, *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*, Ph.D. thesis, California Institute of Technology, 2000.

[PF16] Brian Paden and Emilio Frazzoli, *A generalized label correcting method for optimal kinodynamic motion planning*, Algorithmic Foundations of Robotics XII (2016), Available at: http://www.wafr.org/program.html#detailedprogram.

[PPP02] Stephen Prajna, Antonis Papachristodoulou, and Pablo A Parrilo, *Introducing SOSTOOLS: A general purpose sum-of-squares programming solver*, Conference on Decision and Control (CDC), vol. 1, IEEE, 2002, pp. 741–746.

[PVF16] Brian Paden, Valerio Varricchio, and Emilio Frazzoli, *Sum-of-squares heuristic synthesis for kinodynamic motion planning*, Available at: `https://github.com/bapaden/Sum_of_Squares_Admissible_Heuristics/releases`.

[PVF17] Brian Paden, Valerio Varricchio, and Emilio Frazzoli, *Verification and synthesis of admissible heuristics for kinodynamic motion planning*, IEEE Robotics and Automation Letters **2** (2017), no. 2, 648–655.

[Stu99] Jos F Sturm, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimization methods and software **11** (1999), no. 1-4, 625–653.

[TTT99] Kim-Chuan Toh, Michael J Todd, and Reha H Tutuncu, *SDPT3-a MATLAB software package for semidefinite programming, version 1.3*, Optimization methods and software **11** (1999), no. 1-4, 545–581.

Chapter 7

Conclusion

This thesis presented a direct, forward search algorithm for approximate optimal kinodynamic motion planning called the generalized label correcting method. The contribution of this method to robotics is that it is resolution complete, does not rely on abstract subroutines which may be unavailable in practice, and the assumptions placed on problem data are readily verified.

In Chapter 3, techniques from topology and functional analysis were used to establish that the set of solutions to the problem is open and the performance objective is continuous. These properties were the foundation for the numerical approximation of the problem in Chapter 4, as well as the derivation of the generalized label correcting conditions in Chapter 5. The generalized label correcting method is a slight variation on canonical label correcting methods, making the method familiar and easily implemented. Numerical experiments tested the algorithm on a wide variety of motion planning problems demonstrating that, in addition to the theoretical guarantees, the method performs well in practice.

Like a best-first search, the generalized label correcting method can utilize an admissible heuristic to reduce the running time of the algorithm. This motivated the study of admissible heuristics for kinodynamic motion planning in Chapter 6, which resulted in tools for the verification and synthesis of admissible heuristics. The main result of Chapter 6 was an admissibility condition which is readily verified for a candidate heuristic and which characterizes a convex set of admissible heuristics. Additionally, the synthesis of an admissible heuristic was formulated as an infinite-dimensional linear program whose feasible set consisted of admissible heuristics and whose exact solution is equal to the optimal cost-to-go. Sum-of-squares programming was then proposed as a relaxation of this infinite-dimensional convex program which provides the best provably admissible heuristic within a finite-dimensional subspace of polynomials.

7.1 Extensions and Improvements

There are several directions of future research that will improve the utility of these methods. First and foremost, an anytime variation is needed to maximize effectiveness

in real-time applications. Such a variation was omitted from this thesis to avoid obscuring the simplicity of the method. Secondly, the partitioning of the state space in Section 5.1 is admittedly simple and can be improved with additional information about the dynamical system. For example, the Ball-Box theorem from differential geometry can be leveraged to determine the most appropriate asymptotic scaling of the partition.

One limitation of the generalized label correcting algorithm as it was presented is that it was constructed under very weak assumptions. While this is an attractive property of the method, it leaves information in more structured problem instances unused. It is possible that, with added assumptions, sharper pruning conditions can be derived by similar arguments to those presented in Chapter 5.

Equipped with the understanding of the set of feasible trajectories, an exciting direction for future research is the application of this approach to more complex logical specifications. Simply adding specifications with boolean logic would greatly increase the expressiveness of the generalized label correcting method. However, more sophisticated modal logics, such as linear temporal logic, could be considered as well.

Appendix A

Review of Set Theory

A *set* is a collection of logical variables which are called the *elements* of a set. A set is entirely determined by its elements. Shorthand for the sentence "x is an element of S" is "$x \in S$".

A.1   Notation

Sets can be defined implicitly by the variables x satisfying a logical formula $\mathcal{P}(x)$. Such a set is denoted $\{x : \mathcal{P}(x)\}$. Complex formulas can be constructed from atomic formulas using the logical connectives; "and ($\wedge$)", "or ($\vee$)", "not ($\neg$)", "implies ($\Leftarrow$)", and "equivalent to ($\Leftrightarrow$)"; and the quantifiers "there exists ($\exists$)", and "for all ($\forall$)".

Let S and U be sets. The set S is said to be s *subset* of U, denoted $S \subset U$, if every element of S is also an element of U:

$$(S \subset U) \Leftrightarrow (x \in S \Rightarrow x \in U). \tag{A.1.1}$$

The set S is *equal* to U if S is a subset of U and U is a subset of S:

$$(S = U) \Leftrightarrow (S \subset U \wedge U \subset S). \tag{A.1.2}$$

It is often useful to consider the set of all subsets of a set S. This set, denoted 2^S, is called the *power set* of S,

$$2^S = \{U : U \subset S\}. \tag{A.1.3}$$

A set which contains no elements is called the *empty set*, denoted $\emptyset$.

A.2   Operations on Sets

The *complement* of S, denoted S^c, is the set whose elements are not elements of S:

$$S^c = \{x : \neg(x \in S)\}. \tag{A.2.1}$$

The *union* of S and U, denoted $S \cup U$, is the set whose elements are an element of S or U:

$$S \cup U = \{x : x \in S \lor x \in U\}. \tag{A.2.2}$$

The *intersection* between S and U, denoted $S \cap U$, is the set whose elements are elements of S and U:

$$S \cap U = \{x : x \in S \land x \in U\}. \tag{A.2.3}$$

Two sets S and U are said to be *disjoint* if their intersection is empty, $S \cap U = \emptyset$. The *exclusion* of U from S, denoted $S \setminus U$, is the set whose elements belong to S and do not belong to U:

$$S \setminus U = \{x : x \in S \land (\neg(x \in U))\}. \tag{A.2.4}$$

There are a number of useful identities that follow directly from these operations. These are: associativity of intersection and union operations, $(A \cup B) \cup C = A \cup (B \cup C)$ and $(A \cap B) \cap C = A \cap (B \cap C)$; commutativity of intersection and union operations, $A \cup B = B \cup A$ and $A \cap B = B \cap A$; distributivity $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ and $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$; DeMorgan's law, $(A \cup B)^c = A^c \cap B^c$; and $A \setminus B = A \cap B^c$.

A *cover* of a set S is a collection of sets whose union contains S as a subset. That is, $\{U_i\}_{i \in \mathcal{I}}$ is a cover of S if

$$S \subset \bigcup_{i \in \mathcal{I}} U_i. \tag{A.2.5}$$

If $\{U_i\}_{i \in \mathcal{I}}$ is a cover of S, then a collection of subsets $\{U_i\}_{i \in \mathcal{J}}$ is a *subcover* of $\{U_i\}_{i \in \mathcal{I}}$ if $\mathcal{J} \subset \mathcal{I}$ and

$$S \subset \bigcup_{i \in \mathcal{J}} U_i. \tag{A.2.6}$$

A cover P of S composed of pair-wise disjoint subsets is called a *partition* of S.

A.3 Binary Relations

Given two sets U and W, a *binary relation* R between U and W is a subset of $U \times W$ identifying the elements of U related to elements of W. If $u \in U$ is related to $w \in W$, then $(u, w) \in R$ This is sometimes alternatively denoted uRw. As an example, suppose S is a set of books. A binary relation R on $S \times S$ could be ordered pairs of books where the first was published before the second. Then for $x, y \in S$, if $(x, y) \in R$, the statement "x was published before y" is true.

Two important concepts are the *inverse relation* and *composition relation*. For any relation $R \subset U \times W$, the inverse relation is defined

$$R^{-1} := \{(w, u) \in W \times U : (u, w) \in R\}. \tag{A.3.1}$$

For two relations $R_1 \subset V \times U$ and $R_2 \subset U \times W$, the composition relation on $V \times W$ is defined

$$R_2 \circ R_1 := \{(v, w) \in V \times W : (\exists u)\,((v, u) \in R_1 \land (u, w) \in R_2)\}. \tag{A.3.2}$$

A.3.1 Functions

A *function* f from X into Y, usually denoted $f : X \to Y$, is a binary relation in $X \times Y$ with the properties,

1. For all $x \in X$ there exists a $y \in Y$ such that $(x, y) \in f$.

2. If $(x, y_1) \in f$ and $(x, y_2) \in f$, then $y_1 = y_2$.

In light of the uniqueness property, we write $f(x)$ to denote the element in Y associated to x. The set X is referred to as the *domain* of f and Y is referred to as the *codomain*.

The inverse relation f^{-1} of a function f is not necessarily a function. When the inverse relation of a function is itself a function, the original function is said to be *invertible*.

The following are a few useful properties relating how basic set operations are preserved between the domain and codomain of a function. Let f be a function from X into Y with $A_i \subset X$ and $B_i \subset Y$.

1. $f(A_1 \cup A_2) = f(A_1) \cup f(A_2)$,

2. $f(A_1 \cap A_2) \subset f(A_1) \cap f(A_2)$,

3. $f(A_1) \setminus f(A_2) \subset f(A_1 \setminus A_2)$

4. $f^{-1}(B_1 \cup B_2) = f^{-1}(B_1) \cup f^{-1}(B_2)$,

5. $f^{-1}(B_1 \cap B_2) = f^{-1}(B_1) \cap f^{-1}(B_2)$,

6. $f^{-1}(B_2 \setminus B_1) = f^{-1}(B_2) \setminus f^{-1}(B_1)$

For any set S, a *sequence* on S is a function in $\mathbb{N} \times S$ or $\{1, 2, ..., n\} \times S$. While one would normally write $s(i)$ for the element (i, s) in the sequence, the notation $\{s_i\}_{i \in \mathbb{N}}$ is usually used. When the index set is clear, this is abbreviated $\{s_i\}$. A *subsequence* of s_j of s_i is a function from a subset of the natural numbers into S which is equal to s_i when $i = j$.

If f is a function from X into X, then a *fixed point* of f is an element $x \in X$ satisfying $(x, x) \in f$ (usually written $f(x) = x$).

A.3.2 Orders

An *order* on a set X is a binary relation $R \subset X \times X$ with the properties,

1. $(x, x) \in R$.

2. If $(x, y) \in R$ and $(y, x) \in R$, then $x = y$.

3. If $(x, y) \in R$ and $(y, z) \in R$, then $(x, z) \in R$.

4. $(x, y) \in R$ or $(y, x) \in R$ for all $x, y \in X$.

A relation in $X \times X$ is a *partial order* if it only satisfies items 1-3. It is a *strict partial order* if it satisfies items 2-3 and the negation of item 1. The notation for an order $R \subset X \times X$ is usually denoted $x \leq y$ for $(x, y) \in R$.

Let $(X, \leq)$ be an ordered set. A *lower bound* l on a subset S of X is an element of X such that $l \leq s$ for all $s \in S$. Similarly, an *upper bound* u on a subset S of X is an element of X such that $s \leq u$ for all $s \in S$. Now let L_S be the set of lower bounds of S, and U_S be the set of upper bounds of S. If L_S is nonempty, then S is said to be bounded from below. Similarly, if U_S is nonempty, then S is said to be bounded from above. The ordered set $(X, \leq)$ is said to have the *greatest lower bound property* if, for every nonempty S that is bounded from below, there exists a lower bound $l^* \in X$ of S which is also an upper bound of L_S. We call l^* a greatest lower bound or *infimum*. Similarly, the ordered set $(X, \leq)$ is said to have the *least upper bound property* if, for every nonempty S that is bounded from above, there exists an upper bound $u^* \in X$ of S which is also a lower bound of U_S. We call u^* a least upper bound or *supremum*.

A direct consequence of these definitions is that an ordered set has the greatest lower bound property if and only if it has the least upper bound property. Additionally, if an ordered set admits an infimum/supremum it is unique.

If a subset of an ordered set contains its infimum/supremum, then we say that the set admits a minimum/maximum element.

A.3.3 Equivalence Relations

An *equivalence relation* on a set X is a binary relation $R \subset X \times X$ with the properties,

1. $(x, x) \in R$ for all $x \in X$.

2. $(x, y) \in R$ if and only if $(y, x) \in R$.

3. If $(x, y) \in R$ and $(y, z) \in R$, then $(x, z) \in R$.

The notation for an equivalence relation is usually denoted $x \sim y$ if $(x, y) \in R$. The *equivalence class* of an element $z \in X$ is the set of all $x \in X$ such that $z \sim x$. This is denoted

$$[z] = \{x \in X : z \sim x\}. \tag{A.3.3}$$

Clearly, the set of all equivalence classes is a cover of X since

$$\bigcup_{x \in X} \{x\} \subset \bigcup_{x \in X} [x]. \tag{A.3.4}$$

Additionally, if $[x]$ and $[y]$ are equivalence classes, either $[x] = [y]$ or $[x] \cap [y] = \emptyset$. Thus, every equivalence relation induces a partition of X composed of the equivalence classes of the relation. Conversely, every partition P of X induces an equivalence relation where $x \sim y$ if $x, y \in P$.

Appendix B

Review of Mathematical Analysis

This appendix on is based on personal notes from the following sources [Kel75, Mun00, Rud64].

B.1 Topological Spaces

Equipping sets with a topology makes it possible to discuss continuity of functions, limits of sequences, compactness, and other useful properties.

A set X together with a collection of subsets $\mathcal{T}_X$ is called a *topological space* on X if

TS1 $\emptyset \in \mathcal{T}_X$ and $X \in \mathcal{T}_X$.

TS2 If $S_i \in \mathcal{T}_X$ for every $i \in I$, then $\bigcup_{i \in I} S_i \in \mathcal{T}_X$.

TS3 $U \cap S \in \mathcal{T}_X$ for every $U, S \in \mathcal{T}_X$.

A collection of subsets of X satisfying axioms TS1-TS3 is called a *topology* on X. The elements of a topology are called *open sets*. A set which is the complement of an open set is called a *closed set*. Note that a set can be both open and closed (e.g. $\emptyset$ in any topological space).

The *interior* of a subset S, denoted $int(S)$, is the union of open subsets of S. The *closure* of a subset S, denoted $cl(S)$ is the intersection of closed sets containing S. A *neighborhood* $\mathcal{N}_p$ of a point p in X is an element of $\mathcal{T}_X$ containing p.

A point p in a subset S of X is said to be a *limit point* of S if every neighborhood of p has a nonempty intersection with $S \setminus \{p\}$. A sequence $\{x_i\}$ in X is said to *converge* to a point p in X if, for any neighborhood $\mathcal{N}_p$ of p, there exists an $N \in \mathbb{N}$ such that $n > \mathbb{N}$ implies $x_i \in \mathcal{N}_p$.

If S is a subset of X, then the *subspace topology* $\mathcal{T}_S$ on S is defined by

$$\mathcal{T}_S = \{S \cap U : U \in \mathcal{T}_X\}. \tag{B.1.1}$$

Lemma B.1.1. *If S is an element of $\mathcal{T}_X$, then the subspace topology $\mathcal{T}_S$ is a subset of $\mathcal{T}_X$.*

Proof. If S is an element of $\mathcal{T}_X$ then so is $S \cap U$ for any U in $\mathcal{T}_X$ by TS3. $\square$

It is important to note that if S is not open, elements of the subspace topology defined by S may not be elements of the original topology.

A collection of subsets $\mathcal{B}_X$ of X is called a *base* for a topology if it is a cover of X, and for every $B_i, B_j \in \mathcal{B}_X$, if x is an element of $B_i \cap B_j$, then there exists a $B_k \subset B_i \cap B_j$ in $\mathcal{B}_X$ such that x is an element of B_k. A topological space is *second countable* if it has a countable base.

Lemma B.1.2. *Let $\mathcal{B}_X$ be a base for a topology on X, and let $\mathcal{T}_B$ be the collection of subsets of X satisfying: for every $U \in \mathcal{T}_B$ and $x \in U$, there exists $B \in \mathcal{B}_X$ such that $x \in B$ and $B \subset U$. The collection of subsets $\mathcal{T}_B$ is a topology on X called the topology generated by $\mathcal{B}_X$.*

Proof. We need to show that $\mathcal{T}_B$ is a topology.

(TS1) It is vacuously true that the empty set belongs to $\mathcal{T}_B$. Next, since $\mathcal{B}_X$ is a cover of X, there exists a $B \in \mathcal{B}_X$ such that $x \in B$ for every $x \in X$ and clearly $B \subset X$. Thus, $X \in \mathcal{T}_B$.

(TS2) Let U_i be an element of $\mathcal{T}_B$ for all $i \in \mathcal{I}$ and define

$$U := \bigcup_{i \in \mathcal{I}} U_i. \tag{B.1.2}$$

For each $x \in U$, there is an $i \in \mathcal{I}$ such that $x \in U_i$. Now by construction of $\mathcal{T}_B$, there exists $B \in \mathcal{B}_X$ such that $x \in B$ and $B \subset U_i$. Then $x \in U$ and $B \subset U$ so that $U \in \mathcal{T}_B$.

(TS3) Consider $U_1, U_2 \in \mathcal{T}_B$ and $x \in U_1 \cap U_2$. By construction of $\mathcal{T}_B$, there exists $B_1, B_2 \in \mathcal{B}_X$ such that $x \in B_1 \subset U_1$, and $x \in B_2 \subset U_2$. By definition of a base, there exists B_3 such that $B_3 \subset B_1 \cap B_2$ and $x \in B_3$. Thus, $B_3 \subset U_1 \cap U_2$ which, by construction of $\mathcal{T}_B$, implies $U_1 \cap U_2 \in \mathcal{T}_B$. $\square$

B.1.1 Compactness

A subset of a topological space is said to be *compact* if every open cover has a finite subcover.

Lemma B.1.3. *If S is a closed subset of a compact set C, then S is compact.*

Proof. Let $\{U_i\}_{i \in \mathcal{I}}$ be an open cover of S. Then $\{S^c\} \cup \{U_i\}_{i \in \mathcal{I}}$ is an open cover of C which will have a finite subcover $\{W_i\}_{i \in \mathcal{J}}$. Then $\{W_i\}_{i \in \mathcal{J}} \setminus \{S^c\}$ is a finite subcover of $\{U_i\}_{i \in \mathcal{I}}$. $\square$

Lemma B.1.4. *If C is a compact subset of a topological space, then every sequence $\{c_n\}$ contained in C has a subsequence converging to a point in C.*

Proof. Suppose that the image of the sequence $\{c_n\}$ is finite. Then c_n maps to at least one point p in C an infinite number of times in which case the subsequence whose image is p converges to p. Now suppose $\{c_n\}$ is infinite, and as a point of contradiction that this subset has no limit points. Then it is closed which implies it

is compact as well since a closed subset of a compact set is compact. Then every point in $\{c_n\}$ will have a neighborhood $\mathcal{N}_{c_n}$ containing only the point c_n. This open cover of $\{c_n\}$ cannot have a finite subcover so it is not compact which is a contradiction. Thus, $\{c_n\}$ must have a limit point in C from which a subsequence converging to that limit point can be constructed. $\qquad\square$

It is interesting to note that the converse is also true, but this fact is not required for the arguments in this thesis.

B.1.2   Continuity

A function from a topological space $(X, \mathcal{T}_X)$ into a topological space $(Y, \mathcal{T}_Y)$ is said to be *continuous* if the inverse image of every open subset of Y under f is an open subset of X. That is,

$$f^{-1}(U) \in \mathcal{T}_X \quad \forall U \in \mathcal{T}_Y. \qquad (B.1.3)$$

Alternatively, the function f is continuous on a subset S of X, if f satisfies the above definition of continuity in the subspace topology associated with S. That is,

$$f^{-1}(U) \cap S \in \mathcal{T}_S \qquad \forall U \in \mathcal{T}_Y. \qquad (B.1.4)$$

It is important to point out that is is not necessarily true that $f^{-1}(U) \cap S \in \mathcal{T}_S$ implies $f^{-1}(U) \cap S \in \mathcal{T}_X$ (i.e. the inverse image intersected with S need not be open in $\mathcal{T}_X$).

Lemma B.1.5. *Let f be a continuous function from a topological space $(X, \mathcal{T}_X)$ into a topological space $(Y, \mathcal{T}_Y)$. If C is a compact subset of X, then $f(C)$ is a compact subset of Y.*

Proof. Let $\{U_i\}_{i \in \mathcal{I}}$ be an open cover of $f(C)$. By the definition of continuity and TS2, $\bigcup_{i \in \mathcal{I}} f^{-1}(U_i)$ is an open cover of C. Since C is compact this cover contains a finite subcover $\{f^{-1}(U_i)\}_{i \in \mathcal{J}}$ of C. Then

$$f\left(\bigcup_{i \in \mathcal{J}} f^{-1}(U_i)\right) \subset \bigcup_{i \in \mathcal{J}} f\left(f^{-1}(U_i)\right) \subset \bigcup_{i \in \mathcal{J}} U_i, \qquad (B.1.5)$$

which is an open cover of $f(C)$. Therefore, C is compact. $\qquad\square$

B.2   Metric Spaces

Metric spaces are sets equipped with a distance function that assign a distance between two points in the set. This distance function is a generalization of distance as we encounter it in two or three dimensional space.

A set X together with a function $d_X : X \times X \to [0, \infty)$ is called a *metric space* if

MS1 $0 \leq d_X(x, y)$ for every x, y in X.

95

MS2 $d_X(x,y) = 0$ if and only if $x = y$.

MS3 $d(x,y) = d(y,x)$ for every x,y in X.

MS4 $d(x,z) \leq d(x,y) + d(y,z)$ for every x,y,z in X

A distance function satisfying axioms MS1-MS4 is called *metric* on X. Notice that if (X,d_X) is a metric space and $S \subset X$, then (S,d_X) is a metric space. This is referred to as a *metric subspace*.

The *open ball* of radius r centered at p in a metric space (X,d_X) is defined

$$B_r(p) = \{x \in X : d_X(p,x) < r\}. \tag{B.2.1}$$

Lemma B.2.1. *Every metric space (X,d_X) has an induced topology generated by $\{B_r(x)\}_{x \in X, r > 0}$. This topology is called the metric topology.*

Proof. It is sufficient to show that $\{B_r(x)\}_{x \in X, r > 0}$ is a base for a topology. Clearly, $\{B_r(x)\}_{x \in X, r > 0}$ is a cover of X since $x \in B_r(x)$ for any $r > 0$. Now let x be an element of $B_{r_1}(p_1) \cap B_{r_2}(p_2)$. Denote $d_X(x, p_1)$ by d_1 and $d_X(x, p_2)$ by d_2. Then $B_{(r_1 - d_1)}(x) \subset B_{r_1}(p_1)$ and $B_{(r_2 - d_2)}(x) \subset B_{r_2}(p_2)$. Next, let $r_3 = \min\{r_1 - d_2, r_2 - d_2\}$. Then $B_{r_3}(x) \subset B_{r_1}(p_1) \cap B_{r_2}(p_2)$. Therefore, $\{B_r(x)\}_{x \in X, r > 0}$ is a base for a topology. $\square$

With the induced topology, every metric space inherits the analytical tools available for a topological space.

B.2.1 Continuity in Metric Spaces

Metric spaces gives rise to an alternative interpretation of continuity:

Lemma B.2.2. *Suppose f is a function from a metric space (X, d_X) into (Y, d_Y). If for every $p \in X$ and $\varepsilon > 0$ there exists a $\delta > 0$ (depending on p and ε) such that $d_X(p, x) < \delta$ implies $d_Y(f(p), f(x)) < \varepsilon$, then f is continuous in the metric topology.*

Proof. Let U be an open set in Y with respect to the metric topology. Choose any $p \in f^{-1}(U)$. It will be sufficient to show that p is an interior point of $f^{-1}(U)$. Since U is open, there exists $B_\varepsilon(f(p))$ which is a subset of U. Now suppose that for every $\varepsilon > 0$ there exists a $\delta > 0$ such that $d_X(p, x) < \delta$ implies $d_Y(f(p), f(x)) < \varepsilon$. Then $f^{-1}(B_\varepsilon(f(p))) \subset B_\delta(p)$. Therefore, $B_\delta(p) \subset f^{-1}(U)$ and, p is an interior point of $f^{-1}(U)$. Thus, $f^{-1}(U)$ is open for any open set U which is the definition of continuity. $\square$

The converse is also true making this an alternative definition of continuity.

One can define stronger forms of continuity in metric spaces. A function f from a metric space (X, d_X) into a metric space (Y, d_Y) is said to be *uniformly continuous* if for every $\varepsilon > 0$ there exists a $\delta > 0$ such that $d_X(x_1, x_2) < \delta$ implies $d_Y(f(x_1), f(x_2)) < \varepsilon$. In contrast to the basic form of continuity, the δ in this definition can be independent of the points x_1, x_2 and depends only on ε.

Lemma B.2.3. *A continuous function f from a compact metric space (X, d_X) into a metric space (Y, d_Y) is uniformly continuous.*

Proof. Since f is continuous, for every $\varepsilon > 0$ and $x \in X$, there exists a ball $B_{\delta_x}(x)$ such that $f(B_{\delta_x}(x)) \subset B_\varepsilon(f(x))$. For each $\varepsilon > 0$, the balls $B_{\delta_x/2}(x)$ form an open cover of X. Thus, there is a finite subcover defined by a finite set of points $\{x_1, ..., x_n\}$. Now let $\delta^* = \min\{\delta_{x_1}, ..., \delta_{x_n}\}$. Since this set is finite, it admits a strictly positive minimum. Now choose $p_1, p_2 \in X$ such that $d_X(p_1, p_2) < \delta^*/2$. Then $d_X(p_1, x_m) < \delta^*/2$ for one of the points x_m and by the triangle inequality $d_X(p_2, x_m) < 2\delta^*$. Thus, $p_1, p_2 \in B_{\delta^*}(x_m)$ which implies $d_X(f(p_1), f(p_2)) < \varepsilon$. δ^* is independent of p_1 and p_2 and therefore, f is uniformly continuous. $\square$

An even stronger form of continuity is Lipschitz continuity. A function f from a metric space (X, d_X) into a metric space (Y, d_Y) is said to be *Lipschitz continuous* if there exists a constant L such that

$$d_X(x_1, x_2) < L \cdot d_Y(f(x_1), f(x_2)) \qquad \forall x_1, x_2 \in X. \tag{B.2.2}$$

The function Lip from the set of functions between two metric spaces into $\mathbb{R}$ will be defined as the greatest lower bound on the set of Lipschitz constants of its argument. That is, if f is a function between two metric spaces, then $\mathrm{Lip}(f)$ is the infimum over Lipschitz constants of f.

Lemma B.2.4. *If C and U are disjoint subsets of a metric space (X, d_X), with C compact and U closed, then there exists a $\rho > 0$ such that*

$$\rho < d_X(c, u), \quad \forall c \in C, u \in U. \tag{B.2.3}$$

Proof. As a point of contradiction suppose that the claim is false. Then there exists a sequence $\{c_n\}$ contained in C and a point u in U such that $\lim_{n \to \infty} d(c_n, u) = 0$. Equivalently, $\lim_{n \to \infty} c_n = u$. Since C is a compact set, there is a subsequence $\{c_m\}$ of $\{c_n\}$ which converges to a point c in C. Then by the triangle inequality,

$$\begin{aligned} d(c, u) &\leq \lim_{m \to \infty} d(c_m, c) + \lim_{m \to \infty} d(c_m, u) \\ &= 0 + 0. \end{aligned} \tag{B.2.4}$$

Thus, $c = u$. Since c is a point in C, u is a point in U, and $C \cap U$ is the empty set, we arrive at a contradiction. $\square$

Corollary B.2.1 (to Lemma B.2.4). *If C is a compact subset of an open set S in a metric space (X, d_X), then there exists a $\rho > 0$ such that*

$$B_\rho(c) \subset S, \qquad \forall c \in C. \tag{B.2.5}$$

Proof. Since C is a subset of S, it is disjoint with S^c, and since S is open, S^c is closed. Then by Lemma B.2.4 there is a $\rho > 0$ such that $d_X(c, u) > \rho$ for all c in C and u in S^c. Equivalently, $B_\rho(c) \cap S^c = \emptyset$ which implies $B_\rho(c) \subset S$. $\square$

B.2.2 Complete Metric Spaces

A sequence x_n in a metric space (X, d_X) is said to be a *Cauchy sequence* if for every $\varepsilon > 0$, there exists an N such that $n, m > N$ implies $d_X(x_n, x_m) < \varepsilon$. The metric space (X, d_X) is said to be a *complete metric space* if every Cauchy sequence converges to a point in X.

Lemma B.2.5. *If C is a closed subset of a complete metric space (X, d_X), then (C, d_X) is a complete metric space.*

Proof. Let x_n be a Cauchy sequence in C. Then since X is complete, $\lim_{n \to \infty} x_n = x$ for some $x \in X$. However, x is then a limit point of C and since C is closed, $x \in C$. Thus, (C, d_X) is complete. $\qquad\square$

A function f from a metric space (X, d_X) into (X, d_X) is called a *contraction* if there exists $\gamma \in [0, 1)$ such that

$$d_X(f(x_1), f(x_2)) \leq \gamma d_X(x_1, x_2) \qquad \forall x_1, x_2 \in X. \tag{B.2.6}$$

Theorem B.2.1 (Banach fixed point theorem). *If f is a contraction on a complete metric space (X, d_X), then there exists a unique fixed point of f in X.*

Proof. Let x_0 be any point in X and define the sequence x_n recursively as

$$x_{n+1} = f(x_n). \tag{B.2.7}$$

Then since f is a contraction, there exists $\gamma < 1$ such that

$$d_X(x_{n+1}, x_n) = d_X(f(x_n), f(x_{n-1})) \leq \gamma d_X(x_n, x_{n-1}), \tag{B.2.8}$$

which by induction yields

$$d_X(x_{n+1}, x_n) \leq \gamma^n d_X(x_1, x_0). \tag{B.2.9}$$

Next we show that x_n is a Cauchy sequence. Choose $m > n$ and apply the triangle inequality MS3 to $d_X(x_n, x_m)$,

$$d_X(x_n, x_m) \leq \sum_{i=n+1}^{m} d_X(x_i, x_{i-1}) \tag{B.2.10}$$

Now apply equation B.2.9 to each term in the right hand side of above expression,

$$d_X(x_n, x_m) \leq (\gamma^n + \gamma^n + 1 + ... + \gamma^{m-1}) d_X(x_1, x_0). \tag{B.2.11}$$

Now using the formula for geometric series we obtain

$$\frac{\gamma^n}{1 - \gamma} d_X(x_1, x_2) \tag{B.2.12}$$

which can be made arbitrarily small by choosing n large enough. Thus, x_n is a Cauchy sequence so it must converge to $x^* \in X$ by assumption that (X, d_X) is complete. Then

$$\lim_{n \to \infty} x_n = x^*, \tag{B.2.13}$$

which is equivalent to

$$\lim_{n \to \infty} f(x_{n-1}) = x^*. \tag{B.2.14}$$

Now note that since f is a contraction, it is Lipschitz continuous and therefore

$$\lim_{n \to \infty} f(x_n) = f(x^*). \tag{B.2.15}$$

Thus, $f(x^*) = x^*$ so x^* is a fixed point of f. To prove uniqueness, suppose that x^* and y^* are fixed points of f. Then $d_x(f(x^*), f(y^*)) \leq \gamma d_X(x^*, y^*)$ and $d_x(f(x^*), f(y^*)) = d_x(x^*, y^*)$. The only solution to both the equation and inequality is if $d_X(x^*, y^*) = 0$ in which case $x^* = y^*$ by MS1. $\square$

B.3 Normed Vector Spaces

A *vector space* is an algebraic structure consisting of a set of *vectors* and a field[1] of *scalars*. A vector space is equipped with an associative and commutative addition "+" operation between vectors and a multiplication "·" operation between a scalar and a vector that is distributive over vectors. In addition, there is an additive identity vector, denoted "0", such that $v + 0 = v$ for every vector v. Lastly, there is an additive inverse to every vector v, denoted $-v$ such that $v + (-v) = 0$. This is abbreviated with $v - v$.

A *normed vector space* is a vector space equipped with a function $\| \cdot \|$ mapping vectors into $[0, \infty)$. This function, called a *norm* must satisfy

NS1 $\|0\| = 0$,

NS2 $\|v\| > 0 \qquad \forall v \in \mathcal{V} \setminus \{0\}$,

NS3 $\|\alpha v\| = |\alpha| \|v\| \qquad \forall \alpha \in \mathbb{R}, v \in \mathcal{V}$,

NS4 $\|v + w\| \leq \|v\| + \|w\| \qquad \forall v, w \in \mathcal{V}$.

An example of a norm on the vector space $\mathbb{R}^n$ is the L_p norm for $p = 1, 2, ..., \infty$:

$$\|v\|_p = \left(\sum_{i=1}^{n} v_i^p \right)^{\frac{1}{p}} \tag{B.3.1}$$

Lemma B.3.1. *For every normed vector space* $(\mathcal{V}, \| \cdot \|)$, *there is an induced metric space* $(\mathcal{V}, d_\mathcal{V})$ *with the metric* $d_\mathcal{V}$ *defined by* $d_\mathcal{V}(v, w) = \|v - w\|$.

---

[1]throughout this thesis the only field considered is $\mathbb{R}$.

Proof. We just need to show that MS1-MS4 follow from the definition of the metric and NS1-NS4. Suppose $v = w$. Then $\|v - w\|$ is the same as writing $\|v - v\|$ which is equal to 0 by NS1. Thus, MS1 is satisfied. Suppose $v \neq w$ so that $v - w \neq 0$. Then $\|v - w\| > 0$ by NS2. Thus, MS2 is satisfied. Symmetry of the metric follows from the compatibility of scalar and vector multiplication, $\|v - w\| = |-1| \|(-1) \cdot v - (-1) \cdot w\| = \|w - v\|$. Thus, MS3 is satisfied. Next, the triangle inequality for the metric is just a rearrangement of the triangle inequality for the norm NS4,

$$
\begin{aligned}
\|u - w\| &= \|u + (-w)\| \\
&= \|u - v + v - w\| \qquad\qquad\text{(B.3.2)} \\
&\leq \|u - v\| + \|v - w\|.
\end{aligned}
$$

Thus, MS4 is satisfied. $\qquad\square$

B.3.1   Banach Spaces

A *Banach space* is a normed vector space $(\mathcal{V}, \|\cdot\|)$ with the property that the induced metric space is complete. The $\mathbb{R}^n$ with any of the L_p norms is a Banach space.

Now consider the set of bounded functions $\mathscr{B}$ mapping a compact normed space $(X, \|\cdot\|_X)$ into a Banach space $(Y, \|\cdot\|_Y)$. The following is a norm on $\mathscr{B}$,

$$
\|f\|_\infty := \sup_{x \in X} \|f(x)\|_Y. \qquad\qquad\text{(B.3.3)}
$$

Lemma B.3.2. *The normed space $(\mathscr{B}, \|\cdot\|_\infty)$ is a Banach space.*

Proof. Let f_n be a Cauchy sequence in $(\mathscr{B}, \|\cdot\|_\infty)$. Then since $\|f_n(x) - f_m(x)\|_Y \leq \|f_n - f_m\|_\infty$ we have that $f_n(x)$ is a Cauchy sequence in $(Y, \|\cdot\|_Y)$ for every $x \in X$. Since Y is complete, a point-wise limit of f_n exists which we denote f. It remains to show that the limit f is bounded. While not all Cauchy sequences converge (if the space is not complete) they are always bounded. There exists an M such that $\|f_n\| < M$ for all n. Therefore, $\|f_n(x)\|_Y \leq \|f_n\|_\infty < M$ for all $n \in \mathbb{N}$ and $x \in X$. Thus, $\|f(x)\|_Y < M$ which implies f is a bounded function. Thus, the Cauchy sequence converges to $f \in \mathscr{B}$. $\qquad\square$

Now let $\mathscr{C}$ be the subset of continuous functions in $\mathscr{B}$.

Lemma B.3.3. $\mathscr{C}$ *is a closed subset of the Banach space $(\mathscr{B}, \|\cdot\|_\infty)$.*

Proof. Let f be a limit point of $\mathscr{C}$. Then there exists $f_n \in \mathscr{C}$ which converges to $f \in \mathscr{B}$. For each $\varepsilon > 0$ there exists $N > 0$ such that $\|f_n - f\|_\infty < \varepsilon$ for all $n > N$. Since f_n is continuous on a compact set X, it is uniformly continuous by Lemma B.2.3. Thus, there exists $\delta > 0$ such that $\|f_n(x_1) - f_n(x_2)\|_Y < \varepsilon$ for all $n > N$ and $x_1, x_2 \in X$ satisfying $\|x_1 - x_2\|_X < \delta$.

Next, we use the triangle inequality to show that f is continuous and is therefore contained in $\mathscr{C}$.

$$
\begin{aligned}
\|f(x_1) - f(x_2)\|_Y \quad \leq \quad & \|f(x_1) - f_n(x_1)\|_Y \\
+ \quad & \|f_n(x_1) - f_n(x_2)\|_Y \\
+ \quad & \|f_n(x_2) - f(x_2)\|_Y.
\end{aligned}
\tag{B.3.4}
$$

Observe that $\|f(x_1) - f_n(x_1)\|_Y$ and $\|f_n(x_2) - f(x_2)\|_Y$ are each less than $\|f - f_n\|_\infty$ which is less than ε. Then since f_n is uniformly continuous we were able to choose x_1, x_2 sufficiently close together so that $\|f_n(x_1) - f_n(x_2)\|_Y$ was also less than ε. Thus,

$$
\|f(x_1) - f(x_2)\|_Y \leq 3\varepsilon,
\tag{B.3.5}
$$

which implies that f is continuous and is therefore contained in $\mathscr{C}$. Since f was an arbitrary limit point of $\mathscr{C}$ we conclude that $\mathscr{C}$ is closed. $\qquad\square$

Theorem B.3.1. *The normed space $(\mathscr{C}, \|\cdot\|_\infty)$ is a Banach space.*

Proof. The normed space $(\mathscr{B}, \|\cdot\|_\infty)$ is a Banach space by Lemma B.3.2 and by Lemma B.3.3, $\mathscr{C}$ is a closed subset of $\mathscr{B}$. If follows from Lemma B.2.5 that $(\mathscr{C}, \|\cdot\|_\infty)$ is a Banach space. $\qquad\square$

Chapter References

[Kel75] John L Kelley, *General topology*, vol. 27, Springer Science & Business Media, 1975.

[Mun00] James R Munkres, *Topology*, Prentice Hall, 2000.

[Rud64] Walter Rudin, *Principles of mathematical analysis*, vol. 3, McGraw-Hill New York, 1964.

Appendix C

Analysis of Functions and Integration

The content of this appendix is based on personal notes compiled from the following sources [CL55, Fel81, KF61, Sal16].

C.1 Measure Spaces

It is difficult to have a precise discussion about dynamical systems without Legesgue integration and measure spaces which utilize the concept of measure for sets. The assignment of measure to subsets of the measure space is the generalization of assigning weight to shapes with volume. While the concept is very intuitive, one subtlety with measure spaces is that not all subsets can be assigned measure in general.

A set X together with a collection $\mathscr{X}$ of subsets, and a function $m : \mathcal{S} \to [0, \infty]$ is called a measure space if

SA1 $\emptyset \in \mathscr{X}$, $X \in \mathscr{X}$.

SA2 $X^c \in \mathscr{X}$ for every X in $\mathscr{X}$.

SA3 $\bigcup_{n \in \mathbb{N}} X_n \in \mathscr{X}$ and $\bigcap_{n \in \mathbb{N}} X_n$ for any sequence X_n contained in $\mathscr{X}$.

SA4 $m(\emptyset) = 0$.

SA5 $m \left(\bigcup_{n \in \mathbb{N}} X_n \right) = \sum_{n \in \mathbb{N}} m(X_n)$ for all sequences X_n in $\mathscr{X}$ such that $X_i \cap X_j = \emptyset$ when $i \neq j$.

The collection of subsets $\mathscr{X}$ is called a *σ-algebra* and the function m is called a *measure*. A set S is said to be *measurable* if S is an element of $\mathscr{X}$. We say that a proposition $p(x)$ is true *almost everywhere* in the measure space $(S, \mathscr{S}, m)$ if the set

$$N = \{x \in S : \neg p(x)\} \tag{C.1.1}$$

is measurable and has measure zero. That is $m(N) = 0$.

Notice that SR1-SR3 are slightly different than the axioms of a topology making measure spaces a similar, but not identical structure. In particular, one could say that a topology is "almost" a σ-algebra. There is a natural σ-algebra associated to

103

every topology. This is known as the *Borel σ-algebra* composed of *Borel sets*. A Borel set is any subset of a topological space that can be constructed from countable intersections, countable unions, and complements of elements of the topology.

With the addition of a measure, a Borel σ-algebra on a topological space becomes a measure space. Borel measure spaces are of little use without a few regularity properties. First, a Borel measure is *locally finite* if every point has a neighborhood with finite measure. Second, a Borel measure space $(X, \mathscr{X}, m_x)$ is *inner regular* if for every $\varepsilon > 0$ and $S \in \mathscr{X}$, there exists a compact $K \subset S$ such that $m_X(S \setminus K) < \varepsilon$.

A function f from a measure space $(X, \mathscr{X}, m_X)$ into a measure space $(Y, \mathscr{Y}, m_Y)$ is said to be a *measurable function* if the preimage of every measurable set in Y is a measurable set in X. That is

$$f^{-1}(U) \in \mathscr{X}, \qquad \forall U \in \mathscr{Y}. \tag{C.1.2}$$

Note that this is identical to the definition of continuity in a topological space if we replace the term "measurable" with "open". In light of the similarity between measure spaces and topological spaces, one might guess that a measurable function between Borel σ-algebra is "almost" a continuous function with respect to the topologies generating the algebras. This is indeed the case and this beautiful connection between measure theory and topology is due to Lusin [Lus12].

Theorem C.1.1 (Lusin). *Let f be a measurable function from a locally finite, inner regular Borel measure space $(X, \mathscr{X}, m_X)$ into a Borel measure space $(Y, \mathscr{Y}, m_Y)$ generated by a second countable topology. For every $\varepsilon > 0$ and X in $\mathscr{X}$, there is a subset of V of X such that*

  1. f is continuous on $X \setminus V$, and

  2. $m_X(V) < \varepsilon$.

There are many proofs of this theorem. This presentation of the proof is an expanded version of Feldman's [Fel81] with added detail.

Proof. Since $\mathcal{T}_Y$ is second countable, it has a countable base $\{W_i\}_{i \in \mathbb{N}}$. By the inner regularity of m_X, there exists open sets U_i in $\mathcal{T}_X$ such that $f^{-1}(W_i) \subset U_i$ and $m_X(U_i \setminus f^{-1}(W_i)) < \varepsilon/(2^i)$ for each $i \in \mathbb{N}$. Now define the set

$$V := \bigcup_{i \in \mathbb{N}} U_i \setminus f^{-1}(W_i). \tag{C.1.3}$$

It follows from Axiom SA5 of a measure space that m_X is sub-additive so that

$$m_X(V) \leq \sum_{i \in \mathbb{N}} m_X(U_i \setminus f^{-1}(W_i)) \tag{C.1.4}$$

Then from the construction of the sets U we have

$$m_X(V) \leq \sum_{i \in \mathbb{N}} \frac{\varepsilon}{2^i} = \varepsilon. \tag{C.1.5}$$

The next step in the proof is to show that $f^{-1}(W_i) \setminus V = U_i \setminus V$. First, it is an immediate consequence of the construction of U_i that

$$f^{-1}(W_i) \setminus V \subset U_i \setminus V. \tag{C.1.6}$$

The reverse inclusion is derived as follows[1]:

$$
\begin{aligned}
U_i \setminus V &\subset U_i \setminus (U_i \setminus f^{-1}(W_i)) \\
&= U_i \cap (U_i \setminus f^{-1}(W_i))^c \\
&= U_i \cap (U_i^c \cup f^{-1}(W_i)) \\
&= U_i \cap f^{-1}(W_i) \\
&= f^{-1}(W_i).
\end{aligned}
\tag{C.1.7}
$$

Note that the last step follows from the construction of U_i. Intersecting the left and right hand sides of C.1.7 yields the desired reverse inclusion. Combining the above two equations, we conclude that

$$U_i \setminus V = f^{-1}(W_i) \setminus V. \tag{C.1.8}$$

Next, we use (C.1.8) to prove that f is continuous on $X \setminus V$. Let W be an open subset of Y such that $f^{-1}(W) \cap V = \emptyset$. The set W can be written as a union of elements of the base W_i

$$W = \bigcup_{i \in \mathcal{I}} W_i \tag{C.1.9}$$

Then

$$
\begin{aligned}
f^{-1}(W) \setminus V &= f^{-1}(\bigcup_{i \in \mathcal{I}} W_i) \setminus V \\
&= \left(\bigcup_{i \in \mathcal{I}} f^{-1}(W_i)\right) \setminus V \\
&= \bigcup_{i \in \mathcal{I}} (f^{-1}(W_i) \setminus V) \\
&= \bigcup_{i \in \mathcal{I}} (U_i \setminus V) \\
&= \left(\bigcup_{i \in \mathcal{I}} U_i\right) \setminus V.
\end{aligned}
\tag{C.1.10}
$$

Since U_i are open in $\mathcal{T}_X$, the inverse image $\left(\bigcup_{i \in \mathcal{I}} U_i\right) \setminus V$ is open in the subspace topology induced by $X \setminus V$. Therefore, f is continuous on $X \setminus V$ which concludes the proof. $\square$

---

[1]It may be helpful to review some the elementary identities of set arithmetic in Appendix A.

C.1.1 Lebesgue Integration

Let $(X, \mathscr{X}, m_X)$ be a measure space. The *indicator function* $I_S : X \to \{0, 1\}$ on a measurable set S is defined as follows:

$$I_S(x) = \begin{cases} 1, & x \in S, \\ 0, & \neg(x \in S). \end{cases} \tag{C.1.11}$$

A *simple function* $s : X \to [0, \infty)$ has the following form

$$s(x) = \sum_{i=1}^{n} c_i I_{W_i}(x), \qquad c_i \geq 0, \, W_i \in \mathscr{X}. \tag{C.1.12}$$

Now let $\mathcal{S}$ be the set of simple functions which are point-wise less than or equal to a nonnegative measurable function $f : X \to [0, \infty)$. The *Lebesgue integral* of f is defined by

$$\int_X f(x) \, m_X(dx) = \sup_{s \in \mathcal{S}} \sum_{i=1}^{n} c_i m_X(W_i). \tag{C.1.13}$$

The Lebesgue integral of a general real-valued measurable function is defined by

$$\int_X f(x) \, m_X(dx) = \int_X \max\{f(x), 0\} \, m_X(dx) - \int_X \max\{-f(x), 0\} \, m_X(dx), \tag{C.1.14}$$

where the two nonnegative functions on the right hand side are defined as in (C.1.13). A measurable function f is said to be Lebesgue integrable if

$$\int_X |f(x)| \, m_X(dx) < \infty. \tag{C.1.15}$$

The Lebesgue integral has the following properties,

1. If f, g are integrable and α, β are real numbers, then

$$\int_X \alpha f(x) + \beta g(x) m_X(dx) = \alpha \int_X f(x) m_X(dx) + \beta \int_X g(x) m_X(dx) \tag{C.1.16}$$

2. If f, g are integrable and $f(x) \leq g(x)$ for all x in X, then

$$\int_X f(x) m_X(dx) \leq \int_X g(x) m_X(dx). \tag{C.1.17}$$

The set of integrable functions denoted $\mathcal{L}^1(X, \mathscr{X}, m_X)$ is a vector space. A semi-norm on this space is given by

$$\|f\|_{\mathcal{L}^1} = \int_X |f(x)| \, m_X(dx). \tag{C.1.18}$$

The vector space denoted $L^1(X, \mathscr{X}, m_X)$ refers to the quotient space of $\mathcal{L}^1(X, \mathscr{X}, m_X)$ with the Kernel of its semi-norm making the quotient space a proper normed space. In fact, it is a Banach space.

C.2   Integral Equations (Dynamical System Models)

Integral equations of the form

$$x(\tau) = x_0 + \int_{[t_0, \tau]} f(x(t), t)\, \mu(dt), \tag{C.2.1}$$

appear frequently in engineering and the sciences when modeling various phenomena. The function $f : \mathbb{R}^n \times \mathbb{R}$ represents a model of some dynamical phenomenon, and the function $x : [t_0, t_f] \to \mathbb{R}^n$ describes the evolution of a system parameterized by a vector in $\mathbb{R}^n$. For example, the laws of classical mechanics (pretty important for robotics) are stated as equations in the form of (C.2.1) or the analogous differential form.

It is therefore important to be confident that

1. this equation has a solution,

2. this solution is unique,

3. and the solution is continuous.

If there were instances with no solution then the framework would have little scientific value for modeling dynamical phenomena. If the second point failed then the model would not be useful for predicting the future state of the system. Lastly, if the solution to (C.2.1) were not continuous then solutions would permit instantaneous changes in the parameterization of a system. Taking these points into consideration highlights the importance and elegance of the fundamental existence-uniqueness theorem stated next.

Theorem C.2.1. *Let f be Lipschitz continuous in its first argument on $\mathbb{R}^n$ and measurable in its second argument. Then there exists a unique solution $x : [t_0, t_f] \to \mathbb{R}^n$ in $(\mathscr{C}, \| \cdot \|_\infty)$.*

Proof. The proof technique is known as the Picard iteration. Let L denote a Lipschitz constant for f. Consider the mapping from one function $x : [t_0, t_0 + 1/(2L)] \to \mathbb{R}^n$ to another function $y : [t_0, t_0 + 1/(2L)] \to \mathbb{R}^n$ given by

$$y(\tau) = \int_{[t_0, \tau]} f(x(t), t)\, \mu(dt), \qquad \tau \in [t_0, t_0 + 1/(2L)]. \tag{C.2.2}$$

Observe that this mapping is well defined for all $x \in \mathscr{C}$ since the assumptions of the theorem are sufficient for the integral in (C.2.2) to be finite for each τ. Further, the integral is continuous with respect to τ so that $y \in \mathscr{C}$ as well[2].

---

[2]We have tacitly assumed that μ is the Lebesgue measure on $\mathbb{R}$.

Denote the mapping from x to y by $y = \mathcal{P}(x)$. Notice that x is a fixed point of $\mathcal{P}$ if and only if it is a solution to (C.2.2) with $t_f = 1/(2L)$.

We will show that the Picard iterate $\mathcal{P}$ is a contraction on $\mathscr{C}$ to reach the result. Choose $x_1, x_2 \in \mathscr{C}$ with the domain $[t_0, t_0 + 1/(2L)]$. Then

$$\|[\mathcal{P}(x_1) - \mathcal{P}(x_2)](\tau)\|_2 = \left\| \int_{[0,\tau]} f(x_1(t), t) - f(x_2(t), t)\, \mu(t) \right\|_2$$
$$\leq \int_{[0,\tau]} \|f(x_1(t), t) - f(x_2(t), t)\|_2\, \mu(t). \tag{C.2.3}$$

Then leveraging the Lipschitz continuity of f in its first argument, there exists an L such that

$$\int_{[0,\tau]} \|f(x_1(t), t) - f(x_2(t), t)\|_2\, \mu(t) \leq \int_{[0,\tau]} L \|x_1(t) - x_2(t)\|_2\, \mu(t)$$
$$\leq L \|x_1 - x_2\|_\infty \int_{[0,\tau]} \mu(t)$$
$$\leq L \|x_1 - x_2\|_\infty (1/(2L))$$
$$\leq \tfrac{1}{2} \|x_1 - x_2\|_\infty. \tag{C.2.4}$$

Thus, we have that

$$\|[\mathcal{P}(x_1) - \mathcal{P}(x_2)](\tau)\|_2 \leq \frac{1}{2} \|x_1 - x_2\|_\infty \qquad \forall \tau \in [t_0, t_0 + 1/(2L)], \tag{C.2.5}$$

which is equivalent to

$$\|\mathcal{P}(x_1) - \mathcal{P}(x_2)\|_\infty \leq \frac{1}{2} \|x_1 - x_2\|_\infty. \tag{C.2.6}$$

Therefore, the Picard iterate is a contraction on the interval $[t_0, t_0 + 1/(2L)]$. By Theorem B.2.1 there exists a unique fixed point to the Picard iterate, and therefore there is a unique solution x_1^* to (C.2.1) on the interval $[t_0, t_0 + 1/(2L)]$. This solution necessarily satisfies $x_1^*(t_0) = x_0$.

Now consider solutions on the time interval $[t_0 + 1/(2L), t_0 + 1/L]$ with initial condition $x_1^*(t_0 + 1/(2L))$, replacing x_0 in (C.2.1). By the same argument, there exists a unique solution x_2^* on this time interval with $x_1^*(t_0 + 1/(2L)) = x_2^*(t_0 + 1/(2L))$. Combining these observations, there is necessarily a unique continuous solution on the combined interval $[t_0, t_0 + 1/L]$. A trivial induction argument leads to the conclusion that there is a unique continuous solution on any finite time interval $[t_0, t_f]$. $\qquad \square$

Next we show that if the dynamic model is bounded by M as in Chapter 2, then the solutions are not only continuous, but Lipschitz continuous with Lipschitz constant M.

Corollary C.2.1. *If, in addition to the assumptions of Theorem 2.1.1, the function f is bounded by M; then there is a unique Lipschitz continuous solution to (C.2.1) with Lipschitz constant M.*

Proof. Let x be a solution to (C.2.1) on a finite time interval. Then for two times $t_1 < t_2$ in this time interval,

$$\begin{aligned}
\|x(t_2) - x(t_1)\|_2 &= \left\|\int_{[t_0, t_2]} f(x(t), t)\, \mu(dt) - \int_{[t_0, t_1]} f(x(t), t)\, \mu(dt)\right\|_2 \\
&= \left\|\int_{[t_1, t_2]} f(x(t), t)\, \mu(dt)\right\|_2 \\
&\le \int_{[t_1, t_2]} \|f(x(t), t)\|_2\, \mu(dt) \\
&\le M|t_2 - t_1|
\end{aligned} \tag{C.2.7}$$

Thus, x is Lipschitz continuous with constant M. $\qquad\square$

Lemma C.2.1 (Gronwall's Inequality). *Suppose z is a continuous function from the interval $[0, T]$ into $\mathbb{R}$. If*

$$z(t) \le z_0 + \int_{[0,t]} \lambda z(\tau)\, \mu(d\tau) \tag{C.2.8}$$

for $z_0, \lambda \in \mathbb{R}$, then

$$z(t) \le z_0 e^{\lambda t}. \tag{C.2.9}$$

Proof. Let $y(t) = \int_{[0,t]} \lambda z(\tau)\, d\mu(\tau)$. The assumption that z is continuous implies that y is differentiable. Now let $w(t) = z_0 + y(t) - z(t)$. We have $w(t) \ge 0$. Rewriting the definition of y in terms of $w(t)$ yields,

$$y(t) = \int_{[0,t]} \lambda\left(y(\tau) + z_0 - w(\tau)\right) \mu(d\tau). \tag{C.2.10}$$

Taking the derivative of y yields

$$\frac{d}{dt} y(t) = \lambda y(t) + \lambda z_0 - \lambda w(t). \tag{C.2.11}$$

This is a linear differential equation with solution

$$y(t) = \int_{[0,t]} \lambda e^{\lambda(t-\tau)} \left(z_0 - w(\tau)\right) \mu(d\tau). \tag{C.2.12}$$

Since $w(t) \ge 0$ we have the inequality

$$\begin{aligned}
y(t) &\le z_0 \int_{[0,t]} \lambda e^{\lambda(t-\tau)}\, \mu(d\tau) \\
&= z_0\left(e^{\lambda t} - 1\right) \\
&= z_0 e^{\lambda t} - z_0.
\end{aligned} \tag{C.2.13}$$

Returning to the definition of $y(t)$ we have

$$\begin{aligned}
z(t) &\le z_0 + y(t) \\
&\le z_0 e^{\lambda t},
\end{aligned}$$

which is Gronwall's inequality. □

Gronwall's inequality is best known for its use a a Lemma in proving the continuity of solutions to (C.2.1) with respect to the initial condition x_0. Continuity with respect to the initial conditions adds to the value of the (C.2.1) as a modeling framework for dynamical systems. Without it, an arbitrarily small change in initial conditions could lead to drastically different outcomes.

Theorem C.2.2. *If the assumptions of Theorem 2.1.1 are met, the the terminal state of the solution to (C.2.1) is Lipschitz continuous with respect to the initial condition x_0.*

Proof. Suppose x_1 and x_2 are solutions to two instances of (C.2.1) on the time interval $[t_0, t_f]$ with different initial conditions. Then

$$\|x_1(t_f) - x_2(t_f)\|_2 = \left\| x_1(t_0) - x_2(t_0) + \int_{[t_0, t_f]} f(x_1(t), t) - f(x_2(t), t)\, \mu(dt). \right\|_2$$

$$\leq \|x_1(t_0) - x_2(t_0)\|_2 + \int_{[t_0, t_f]} \|f(x_1(t), t) - f(x_2(t), t)\|_2\, \mu(dt)$$

$$\leq \|x_1(t_0) - x_2(t_0)\|_2 + \int_{[t_0, t_f]} L\, \|x_1(t) - x_2(t)\|_2\, \mu(dt)$$

$$\tag{C.2.14}$$

As a direct consequence of Lemma C.2.1 we have that

$$\|x_1(t_f) - x_2(t_f)\|_2 \leq \|x_1(t_0) - x_2(t_0)\|\, e^{L(t_f - t_0)}. \tag{C.2.15}$$

□

Chapter References

[CL55] Earl A Coddington and Norman Levinson, *Theory of Ordinary Differential Equations*, McGraw-Hill Education, 1955.

[Fel81] Marcus B Feldman, *A proof of Lusin's theorem*, The American Mathematical Monthly **88** (1981), no. 3, 191–192.

[KF61] Andrej Nikolaevič Kolmogorov and Sergej Vasilévič Fomin, *Elements of the theory of functions and functional analysis. vol. 2, measure. the lebesgue integral. hilbert spaces*, Graylock Press, 1961.

[Lus12] Nikolai Lusin, *Sur les propriétés des fonctions mesurables*, CR Acad. Sci. Paris **154** (1912), 1688–1690.

[Sal16] Dietmar Salamon, *Measure and integration*, 2016.

Appendix D

Graph Search Algorithms

This appendix provides an introduction to shortest path problems on graphs and an informed best-first search, also known as the A^* algorithm.

A *graph* is a set V equipped with a binary relation E on $V \times V$. An element of V is called a *vertex* and an element of E is called an *edge*. A *path* $\{p_1, ..., p_n\}$ is a finite sequence in V such that $(p_i, p_{i+1}) \in E$ for each sequential pair in the sequence. A weighted graph is a graph equipped with a function $c : E \to \mathbb{R}$ which assigns a weight or cost to each edge. The *cost* or *weight* of a path $\{p_i\}$ with length n in a weighed graph is, with some abuse of notation, given by

$$c(\{p_i\}) = \sum_{i=1}^{n-1} c((p_i, p_{i+1})) \tag{D.0.1}$$

A *shortest path* from a source $s \in V$ to a destination set $D \subset V$ is a finite path $\{p_1, ..., p_n\}$ with $p_1 = s$, $p_n \in D$, and which has the minimum cost among paths originating from s and terminating in D.

Lemma D.0.1. *A finite, weighted graph with non-negative edge weights either admits a shortest path from $s \in V$ to $D \subset V$ with cost c^*, or has no path from s to D.*

Proof. Suppose there is a path $\{p_i\}$ from s to D. Then the set of paths from s to D is nonempty. Note that for every path that visits a vertex twice, there exists a path with equal or lesser cost that omits the cycle. Therefore, if a path from s to D exists, then there is a nonempty set of paths from s to D with no cycles. A path with no cycles is bounded in length by the number of vertices in the graph N, and the number of paths of length N is bounded by N^N. Thus, there are a finite number of paths with no cycles. Since every real valued function with a finite domain admits a minimum, there is a minimum cost path within this subset which lower bounds all paths from s to D. $\square$

The A^* algorithm is generally the most effective algorithm for single-source shortest path problems. An essential component of the A^* algorithm is an *admissible heuristic*; a function $h : V \to \mathbb{R}$ which estimates the cost of a shortest path from each vertex to D, and never overestimates it. The algorithm incrementally examines

paths originating from s in order of their cost plus the estimated cost to reach D. The algorithm uses a function $\texttt{label} : V \to \mathbb{R} \cup \{\infty\}$ which is recursively redefined at each iteration with the cost of the best known path reaching each vertex; the function initially maps all vertices to ∞ representing that no path from s to any particular vertex is known. A function $\texttt{parent} : V \to V \cup \{\texttt{NULL}\}$ is redefined in each iteration to implicitly build a secondary graph $(V \cup \{\texttt{NULL}\}, \tilde{E})$ of shortest paths from s to every other vertex. The edge set of this graph is

$$\tilde{E} = \bigcup_{v \in V} \{(\texttt{parent}(v), v)\} \tag{D.0.2}$$

The function $\texttt{parent}$ initially maps all vertices to $\texttt{NULL}$ which represents that no path to these vertices is known.

A set Q of vertices, associated to paths from s by the $\texttt{parent}$ function, is operated on by the algorithm with standard set operations together with an operation $\texttt{pop}$ which returns a vertex $v \in Q$ with terminal vertex satisfying

$$\texttt{pop}(Q) \in \operatorname*{argmin}_{w \in Q} \{\texttt{label}(w) + h(w)\} \leq c^* \tag{D.0.3}$$

The last operation that is needed to present the algorithm is a function $\texttt{neighborhood} : V \to 2^E$ which returns the set of edges in the graph whose first component is the argument of the function. That is,

$$\texttt{neighborhood}(w) = \{e \in E : e = (w, v)\} \tag{D.0.4}$$

Algorithm 2 The A* algorithm.
___
1: $Q \leftarrow s$;
2: $\texttt{label}(s) \leftarrow 0$
3: **while** $\neg(Q = \emptyset)$
4:     $v \leftarrow \texttt{pop}(Q)$
5:     **if** $v \in D$
6:         **return** $(v, \texttt{parent})$
7:     $S \leftarrow \texttt{neighbors}(v)$
8:     **for** $w \in S$
9:         **if** $\texttt{label}(v) + \texttt{cost}(v, w) < \texttt{label}(w)$
10:           $\texttt{label}(w) \leftarrow \texttt{label}(v) + \texttt{cost}(v, w)$
11:           $\texttt{parent}(w) \leftarrow v$
12:           $Q \leftarrow Q \cup \{w\}$
13: **return** $\texttt{NO SOLUTION}$
___

In each iteration of the algorithm, an element of Q with greatest merit is removed from Q and assigned to v (line 5). The algorithm will terminate if $v \in D$ since the

path $(v_1, v_2 ..., v_n)$ satisfying

$$v_{i+1} = \texttt{parent}(v_i), \quad v_1 = s, \ v_n = v \in D, \tag{D.0.5}$$

is a path from s to D. If v is not in the destination set, then the algorithm will continue the search by exploring extensions of the subgraph implicitly defined by the $\texttt{parent}$ function (line 7 - line 12). This is accomplished by assigning the neighbors of v to the set S (line 7). Each of the vertices in S can be reached via the path to v together with the edge connecting v to each vertex in S. The cost of these paths is the cost to reach v, accesed by $\texttt{label}(v)$, plus the cost of the edge from v to the vertex in S. If this cost is less than the current label of any of the vertices in S, the function $\texttt{label}$ is updated with that lower cost (line 10). Additionally, the path to this vertex in S is stored by assigning the parent of that vertex to v (line 11). The last step in each iteration is to add the vertices of S whose label was updated to the set Q for examination in future iterations. The algorithm terminates when a path from s to D is found as discussed above, or when the set Q is empty and there are no more options for finding better paths than has already been discovered.

Theorem D.0.1. *On a finite weighted graph with nonnegative edge weights, the A^* algorithm terminates in finite time returning a shortest path if one exists, or returns* $\texttt{NO SOLUTION}$ *if there is no path from s to D.*

Proof. (Finite running time) During the loop (line 3 - line 12), a vertex is inserted into Q if a lower cost path to that vertex than was previously known was discovered. Since the number of paths with cost less than any particular value is finite, the insertion of any vertex into Q can only occur a finite number of times. Together with the finite number of vertices, the total number of vertices inserted into Q during execution is finite. In each iteration, one vertex is removed from Q. Therefore, the algorithm must eventually terminate by returning $\texttt{NO SOLUTION}$ or a vertex and the parent function.

(Optimality of returned path) Let $\{v_1, ..., v_n\}$ be a shortest path from s to D with cost $c(\{v_1, ..., v_n\}) = c^*$. Suppose, as a point of contradiction, that the algorithm does not return a path with cost c^*. Then v_n is never present in Q with $\texttt{label}(v_n) = c(\{v_1, ..., v_n\})$; the reason being that, if it were inserted into Q with this label, then

$$c(\{v_1, ..., v_n\}) + h(v_n) \leq c^*. \tag{D.0.6}$$

Then, prior to termination, $\texttt{pop}(Q)$ would return $v_n \in D$ at which point the algorithm would terminate with a shortest path—contradictory our hypothesis. It follows that v_{n-1} never enters Q with $\texttt{label}(v_{n-1}) = c(\{v_1, ..., v_{n-1}\})$. Similar to the previous step, we would have

$$c(\{v_1, ..., v_{n-1}\}) + h(v_{n-1}) \leq c^*, \tag{D.0.7}$$

so $\texttt{pop}(Q)$ would return v_{n-1} with this label before termination of the algorithm. Since v_n is a neighbor of v_{n-1} this would result in setting $\texttt{label}(v_n)$ to $c(\{v_1, ..., v_{n-1}\}) + c(\{v_{n-1}, v_n\})$ in line 10 which, as was previously established, does not occur. Continuing these deductive steps leads to the conclusion that $\{v_1\}$ must never enter Q with $\texttt{label}(v_1) = c(\{v_1\}) = 0$. This is a contradiction of line 2 of the algorithm. Thus, if

a path from s to D exists, the algorithm must terminate, returning a vertex $v_n \in D$ with $\texttt{label}(v_n) = c^*$. A shortest path from $v_1 = s$ to $v_n \in D$ can then be recovered with the $\texttt{parent}$ function.

In the event that there is no path from s to D, the algorithm will never insert a vertex $v \in D$ into Q since such vertices are always associated to a path to s through the $\texttt{parent}$ function. Since termination of the algorithm was already established, it must terminate with $\texttt{NO SOLUTION}$.

$\square$

Appendix E

Extended Derivations and Proofs

This appendix contains the tedious and less interesting proofs that are necessary for the completeness of this thesis.

E.1 Proof that $d_{\mathcal{U}}$ is a Metric on $\mathcal{U}$

Recall the the proposed metric $d_{\mathcal{U}}$ for the input signal space $\mathcal{U}$ in Chapter 3,

$$d_{\mathcal{U}}(u_1, u_2) := \int_{[0,\min(\tau(u_1),\tau(u_2)]} \|u_1(t) - u_2(t)\|_2 \, \mu(dt) + u_{max}|\tau(u_1) - \tau(u_2)|. \quad \text{(E.1.1)}$$

The following derivation shows that $d_{\mathcal{U}}$ satisfies axioms MS1-MS4 for a metric: Let u_1, u_2, and u_3 denote elements of $\mathcal{U}$.

(MS1) By the nonnegativity of the norms on $\mathbb{R}$ and $\mathbb{R}^n$, the integrand $\|u_1(t) - u_2(t)\|_2$ and $|\tau(u_1) - \tau(u_2)|$ are nonnegative for all times t. Thus,

$$\begin{aligned}
d_{\mathcal{U}}(u_1, u_2) &= \int_{[0,\min\{\tau(u_1),\tau(u_2)\}]} \|u_1(t) - u_2(t)\|_2 \, \mu(dt) + u_{max}|\tau(u_1) - \tau(u_2)| \\
&\leq \int_{[0,\min\{\tau(u_1),\tau(u_2)\}]} (0) \, \mu(dt) + u_{max}(0) \\
&= 0,
\end{aligned}$$

(E.1.2)

so that MS1 is satisfied.

(MS2) Consider two input signals u_1, u_2 which are not equal. Then at least one of the following is true: (i) their time domains differ so that $|\tau(u_1) - \tau(u_2)| > 0$, (ii) there is a subset of $S \subset [0, \min\{\tau(u_1), \tau(u_2)\}]$ on which $u_1(t) \neq u_2(t)$ in which case $\|u_1(t) - u_2(t)\|_2 > 0$. For the latter case, recall that the L_1-spaces are quotient spaces whose elements are the equivalence classes of signals that differ only on sets of measure zero. Thus, S has positive measure. Then,

$$\begin{aligned}
d_{\mathcal{U}}(u_1, u_2) &= \int_{[0,\min\{\tau(u_1),\tau(u_2)\}]} \|u_1(t) - u_2(t)\|_2 \, \mu(dt) + u_{max}|\tau(u_1) - \tau(u_2)| \\
&= \int_S \|u_1(t) - u_2(t)\|_2 \, \mu(dt) + u_{max}|\tau(u_1) - \tau(u_2)| \\
&> 0.
\end{aligned}$$

(E.1.3)

In the case that $u_1 = u_2$, we have $u_1(t) = u_2(t)$ for every t in $[0, \tau(u_1)]$. Additionally,

$\tau(u_1) = \tau(u_2)$ so the distance between u_1 and u_2 is zero when they are equal. Thus, MS2 is satisfied.

(MS3) Next, symmetry of the metric follows from the symmetry of the Euclidean metric from which it is constructed,

$$
\begin{aligned}
d_{\mathcal{U}}(u_1, u_2) &= \int_{[0,\min\{\tau(u_1),\tau(u_2)\}]} \|u_1(t) - u_2(t)\|_2\, \mu(dt) + u_{max}|\tau(u_1) - \tau(u_2)| \\
&= \int_{[0,\min\{\tau(u_2),\tau(u_1)\}]} \|u_2(t) - u_1(t)\|_2\, \mu(dt) + u_{max}|\tau(u_2) - \tau(u_1)| \\
&= d_{\mathcal{U}}(u_2, u_1),
\end{aligned}
\tag{E.1.4}
$$

so that MS3 is satisfied.

(MS4) To simplify the notation of the proof, let $\tau(u_1) \leq \tau(u_2) \leq \tau(u_3)$. To show that triangle inequality holds together with this assumption, for any three points in $\mathcal{U}$ it will need to be shown for all permutations of u_1, u_2, u_3. From the triangle inequality for the Euclidean metric we have,

$$
\begin{aligned}
d_{\mathcal{U}}(u_1, u_3) &= \int_{[0,\tau(u_1)]} \|u_1(t) - u_3(t)\|_2\, \mu(dt) + u_{max}|\tau(u_1) - \tau(u_3)| \\
&\leq \int_{[0,\tau(u_1)]} \|u_1(t) - u_2(t)\|_2 + \|u_2(t) - u_3(t)\|_2\, \mu(dt) \\
&\quad + u_{max}|\tau_2 - \tau_1| + u_{max}|\tau_3 - \tau_2|. \\
&= \int_{[0,\tau(u_1)]} \|u_1(t) - u_2(t)\|_2\, \mu(dt) + u_{max}|\tau(u_2) - \tau(u_1)| \\
&\quad + \int_{[0,\tau(u_1)]} \|u_2(t) - u_3(t)\|_2\, \mu(dt) + u_{max}|\tau(u_3) - \tau(u_2)|. \\
&\leq \int_{[0,\tau(u_1)]} \|u_1(t) - u_2(t)\|_2\, \mu(dt) + u_{max}|\tau(u_2) - \tau(u_1)| \\
&\quad + \int_{[0,\tau(u_2)]} \|u_2(t) - u_3(t)\|_2\, \mu(dt) + u_{max}|\tau(u_3) - \tau(u_2)|. \\
&= d_{\mathcal{U}}(u_1, u_2) + d_{\mathcal{U}}(u_2, u_3)
\end{aligned}
\tag{E.1.5}
$$

Note that in the second step above, the integration in the second term is carried out over a larger domain to arrive at the desired result. The next permutation is nearly identical to the first,

$$
\begin{aligned}
d_{\mathcal{U}}(u_1, u_2) &= \int_{[0,\tau(u_1)]} \|u_1(t) - u_2(t)\|_2\, \mu(dt) + u_{max}|\tau(u_1) - \tau(u_2)| \\
&\leq \int_{[0,\tau(u_1)])} \|u_1(t) - u_3(t)\|_2 + \|u_2(t) - u_3(t)\|_2\, \mu(dt) \\
&\quad + u_{max}|\tau(u_1) - \tau(u_3)| + u_{max}|\tau(u_2) - \tau(u_3)|. \\
&= \int_{[0,\tau(u_1)]} \|u_1(t) - u_3(t)\|_2\, \mu(dt) + u_{max}|\tau(u_1) - \tau(u_3)| \\
&\quad + \int_{[0,\tau(u_1)]} \|u_2(t) - u_3(t)\|_2\, \mu(dt) + u_{max}|\tau(u_2) - \tau(u_3)|. \\
&\leq \int_{[0,\tau(u_1)]} \|u_1(t) - u_3(t)\|_2\, \mu(dt) + u_{max}|\tau(u_1) - \tau(u_3)| \\
&\quad + \int_{[0,\tau(u_2)]} \|u_2(t) - u_3(t)\|_2\, \mu(dt) + u_{max}|\tau(u_2) - \tau(u_3)|. \\
&= d_{\mathcal{U}}(u_1, u_3) + d_{\mathcal{U}}(u_2, u_3)
\end{aligned}
\tag{E.1.6}
$$

Now the third permutation,

$$
\begin{aligned}
d_{\mathcal{U}}(u_2, u_3) &= \int_{[0,\tau(u_2)]} \|u_2(t) - u_3(t)\|_2\, \mu(dt) + u_{max}|\tau(u_2) - \tau(u_3)| \\
&\leq \int_{[0,\tau(u_1)]} \|u_2(t) - u_3(t)\|_2\, \mu(dt) + u_{max}|\tau(u_2) - \tau(u_3)| \\
&\leq \int_{[0,\tau(u_1)]} \|u_1(t) - u_2(t)\|_2\, \mu(dt) + u_{max}|\tau(u_1) - \tau(u_2)| \\
&\quad + \int_{[0,\tau(u_1)]} \|u_1(t) - u_3(t)\|_2\, \mu(dt) + u_{max}|\tau(u_1) - \tau(u_3)| \\
&= d_{\mathcal{U}}(u_1, u_2) + d_{\mathcal{U}}(u_1, u_3).
\end{aligned}
\tag{E.1.7}
$$

The remaining three cases follow trivially from the symmetry property already established. Thus, the function $d_{\mathcal{U}}$ is a metric on the set of input signals $\mathcal{U}$.

E.2   Proof that $d_{\mathcal{X}}$ is a metric on $\mathcal{X}$

Recall the the proposed metric $d_{\mathcal{X}}$ for the input signal space $\mathcal{X}$ in Chapter 3,

$$d_{\mathcal{X}}(x_1, x_2) := \max_{t \in [0, \min\{\tau(x_1), \tau(x_2)\}]} \{\|x_1(t) - x_2(t)\|_2\} + M|\tau(x_1) - \tau(x_2)|. \quad \text{(E.2.1)}$$

The following derivation shows that $d_{\mathcal{X}}$ satisfies axioms MS1-MS4 for a metric: Let x_1, x_2, and x_3 be elements of $\mathcal{X}$.

(MS1) For each $t \in [0, \min\{\tau(x_1), \tau(x_2)\}]$ the term $+M|\tau(x_1) - \tau(x_2)|$ is nonnegative as is $\|x_1(t) - x_2(t)\|_2$. The maximum over a set of nonnegative numbers is nonnegative so $d_{\mathcal{X}}$ is nonnegative.

(MS2) Suppose $d_{\mathcal{X}}(x_1, x_2) = 0$. Then

$$\max_{t \in [0, \min\{\tau(x_1), \tau(x_2)\}]} \{\|x_1(t) - x_2(t)\|_2\} + M|\tau(x_1) - \tau(x_2)| = 0. \quad \text{(E.2.2)}$$

Since the expression is a sum of two nonnegative terms adding to zero, each term is necessarily zero. Therefore, $\tau(x_1) = \tau(x_2)$, and $x_1(t) = x_2(t)$ for all $t \in [0, \tau(x_1)]$ and $t \in [0, \tau(x_2)]$. Equivalently, $x_1 = x_2$. Now suppose $x_1 = x_2$. Then $\tau(x_1) = \tau(x_1)$ and $x_1(t) = x_2(t)$ for every t in $[0, \tau(x_1)]$ and $[0, \tau(x_2)]$. This implies $|\tau(x_1) - \tau(x_1)| = 0$ and $\|x_1(t) - x_2(t)\|_2 = 0$. Therefore, $d_{\mathcal{X}}(x_1, x_2) = 0$.

(MS3) Symmetry of the distance function is inherited from the symmetry of the metrics from which it is constructed,

$$
\begin{aligned}
d_{\mathcal{X}}(x_1, x_2) &= \max_{t \in [0, \min\{\tau(x_1), \tau(x_2)\}]} \{\|x_1(t) - x_2(t)\|\} + M|\tau(x_1) - \tau(x_2)| \\
&= \max_{t \in [0, \min\{\tau(x_2), \tau(x_1)\}]} \{\|x_2(t) - x_1(t)\|\} + M|\tau(x_2) - \tau(x_1)| \quad \text{(E.2.3)} \\
&= d_{\mathcal{X}}(x_2, x_1)
\end{aligned}
$$

(MS4) To simplify the notation, assume $\tau(x_1) \leq \tau(x_2) \leq \tau(x_3)$. To show that triangle inequality holds together with this assumption, the inequality it will have to be shown for all permutations of x_1, x_2, x_3: First,

$$
\begin{aligned}
d_{\mathcal{X}}(x_1, x_3) &= \max_{t \in [0, \tau_1]} \{\|x_1(t) - x_3(t)\|\} + M|\tau_3 - \tau_1| \\
&\leq \max_{t \in [0, \tau_1]} \{\|x_1(t) - x_2(t)\|\} + \max_{t \in [0, \tau_1]} \{\|x_2(t) - x_3(t)\|\} \\
&\quad + M|\tau_2 - \tau_1| + M|\tau_3 - \tau_2|. \\
&\leq \max_{t \in [0, \tau_1]} \{\|x_1(t) - x_2(t)\|\} + \max_{t \in [0, \tau_2]} \{\|x_2(t) - x_3(t)\|\} \quad \text{(E.2.4)} \\
&\quad + M|\tau_2 - \tau_1| + M|\tau_3 - \tau_2|. \\
&= d_{\mathcal{X}}(x_1, x_2) + d_{\mathcal{X}}(x_2, x_3).
\end{aligned}
$$

117

Next,

$$
\begin{aligned}
d_{\mathcal{X}}(x_1, x_2) &= \max_{t \in [0,\tau_1]} \left\{ \|x_1(t) - x_2(t)\| \right\} + M|\tau_3 - \tau_1| \\
&\leq \max_{t \in [0,\tau_1]} \left\{ \|x_1(t) - x_3(t)\| \right\} + \max_{t \in [0,\tau_1]} \left\{ \|x_2(t) - x_3(t)\| \right\} \\
&\quad + M|\tau_1 - \tau_3| + M|\tau_2 - \tau_3|. \\
&\leq \max_{t \in [0,\tau_1]} \left\{ \|x_1(t) - x_2(t)\| \right\} + \max_{t \in [0,\tau_2]} \left\{ \|x_2(t) - x_3(t)\| \right\} \\
&\quad + M|\tau_1 - \tau_3| + M|\tau_2 - \tau_3|. \\
&= d_{\mathcal{X}}(x_1, x_3) + d_{\mathcal{X}}(x_2, x_3)
\end{aligned} \tag{E.2.5}
$$

Then,

$$
\begin{aligned}
d_{\mathcal{X}}(x_2, x_3) &= \max_{t \in [0,\tau_2]} \left\{ \|x_2(t) - x_3(t)\| \right\} + M|\tau_3 - \tau_2| \\
&\leq \max_{t \in [0,\tau_1]} \left\{ \|x_1(t) - x_2(t)\| \right\} + \max_{t \in [0,\tau_1]} \left\{ \|x_1(t) - x_3(t)\| \right\} \\
&\quad + M|\tau_1 - \tau_2| + M|\tau_1 - \tau_3| \\
&= d_{\mathcal{X}}(x_1, x_2) + d_{\mathcal{X}}(x_1, x_3).
\end{aligned} \tag{E.2.6}
$$

The remaining three cases follow trivially from the symmetry property already established.

E.3 Derivation of Inequality (5.4.2)

This section derives the inequality

$$
\sum_{i=1}^{h(R)} \frac{\sqrt{n}}{\eta(R)} e^{\frac{L_f(h(R)-i)}{R}} \leq \frac{R\sqrt{n}}{L_f \eta(R)} \left(e^{L_f h(R)/R} - 1 \right). \tag{E.3.1}
$$

First, we factor out the term $\sqrt{n}/\eta(R)$,

$$
\sum_{i=1}^{h(R)} \frac{\sqrt{n}}{\eta(R)} e^{\frac{L_f(h(R)-i)}{R}} = \frac{\sqrt{n}}{\eta(R)} \sum_{i=1}^{h(R)} e^{\frac{L_f(h(R)-i)}{R}}, \tag{E.3.2}
$$

followed by reversing the order of the summation,

$$
\frac{\sqrt{n}}{\eta(R)} \sum_{i=1}^{h(R)} e^{\frac{L_f(h(R)-i)}{R}} = \frac{\sqrt{n}}{\eta(R)} \sum_{i=0}^{h(R)-1} e^{\frac{L_f i}{R}}. \tag{E.3.3}
$$

Note that for a general nondecreasing function $f : \mathbb{R} \to \mathbb{R}$ we have the inequality

$$
\sum_{i \in \{j, j+1, \dots, k\}} f(i) \leq \int_j^k f(\rho) d\rho, \tag{E.3.4}
$$

since the sum is the left-Riemann sum if the integral. This can be applied to (E.3.3) to obtain

$$
\begin{aligned}
\frac{\sqrt{n}}{\eta(R)} \sum_{i=0}^{h(R)-1} e^{\frac{L_f i}{R}} &\leq \frac{\sqrt{n}}{\eta(R)} \int_0^{h(R)} e^{\frac{L_f}{R}\rho} d\rho \\
&= \frac{R\sqrt{n}}{L_f \eta(R)} \left(e^{\frac{L_f h(R)}{R}} - 1 \right),
\end{aligned}
\tag{E.3.5}
$$

which is the desired inequality.

Index

minimal (partial order), 45

neighborhood, 93
norm, 99
normed vector space, 99

open ball, 96
open set, 93
order, 91

partial order, 92
partition, 90
path (graph), 111
power set, 89
principle of optimality, 48
probabilistic completeness, 16
problem data, 24
problem relaxation, 74

relaxed optimal kinodynamic motion planning problem, 24
resolution complete, 24
resolution completeness, 16
running cost, 23

scalars, 99
second countable topological space, 94
sequence, 91
set exclusion, 90
shortest path (graph), 111
simple function, 106
solution (integral equation), 22
strict partial order, 92
subcover, 90
subsequence, 91
subset, 89
subspace topology, 93
supremum, 92
system map, 23

terminal control input, 22
terminal state, 22
terminal time, 22
topological space, 93
topology, 93
trajectory, 21

uniform continuity, 96
union, 90
upper bound, 92

value function, 67
vector space, 99
vectors, 99
vertex, 111
viscosity solution, 68