3-1-2016

Implications of Motion Planning: Optimality and k-survivability

Yu-Han Lyu
Dartmouth College

IMPLICATIONS OF MOTION PLANNING:

OPTIMALITY AND K-SURVIVABILITY

Dartmouth Computer Science Technical Report TR2016-791 by

Yu-Han Lyu

DARTMOUTH COLLEGE

Hanover, New Hampshire

March 2016

Abstract

We study motion planning problems, finding trajectories that connect two configurations of a system, from two different perspectives: optimality and survivability. For the problem of finding optimal trajectories, we provide a model in which the existence of optimal trajectories is guaranteed, and design an algorithm to find approximately optimal trajectories for a kinematic planar robot within this model. We also design an algorithm to build data structures to represent the configuration space, supporting optimal trajectory queries for any given pair of configurations in an obstructed environment.

We are also interested in planning paths for expendable robots moving in a threat environment. Since robots are expendable, our goal is to ensure a certain number of robots reaching the goal. We consider a new motion planning problem, *maximum k-survivability*: given two points in a stochastic threat environment, find n paths connecting two given points while maximizing the probability that at least k paths reach the goal. Intuitively, a good solution should be diverse to avoid several paths being blocked simultaneously, and paths should be short so that robots can quickly pass through dangerous areas. Finding sets of paths with maximum k-survivability is NP-hard. We design two algorithms: an algorithm that is guaranteed to find an optimal list of paths, and a set of heuristic methods that finds paths with high

k-survivability.

Acknowledgments

First, I would like to acknowledge my advisor, Dr. Balkcom for his support of my research and suggestions on my thesis. I also appreciate all of my committee members for their helpful mentorships of my research.

Moreover, I am thankful for the support from Dartmouth College, especially for the BorrowDirect service at Dartmouth Library, which allows me to borrow hundreds of books from various universities easily, and for the digital library, which lets me enable to accumulate invaluable treasures – thousands of academic papers from numerous sources. I also would like to acknowledge the Department of Computer Science, which provides a wonderful research atmosphere, including world-class faculty members, cutting-edge research projects, and hard-working students.

My gratitude extends to all members of the robotics lab at Dartmouth College, including Weifu Wang, Yinan Chang, and Yining Chen, and to all my friends at Dartmouth College for many hours of insightful discussions.

Finally, I deeply thank my wife and my family members. Without their support, I cannot finish my Ph.D. degree.

Contents

Abstract ii

Acknowledgements iv

1 Introduction **1**

2 Related work **7**

2.1 Complete motion planning 7

2.2 Resolution complete motion planning 8

2.3 Probabilistically complete motion planning 10

2.3.1 Asymptotically optimal motion planning 10

2.4 Topological analysis in motion planning 11

2.5 Nonholonomic motion planning 11

2.5.1 Differential geometry 12

2.5.2 Control theory 12

3 Single query for optimal trajectories **14**

3.1 Introduction 14

3.1.1 Model and notation 19

3.1.2 Related work 20

3.2 Mathematical background 27

3.2.1 Blatt's indifference principle 27

3.2.2 Karush-Kuhn-Tucker conditions 29

3.3 Necessary conditions for optimal trajectories 30

3.3.1 Control line trajectories 32

3.3.2 Whirl trajectories 35

3.3.3 Taxonomy of optimal trajectories 37

3.4 The bench mover's problem 38

3.4.1 Model and trajectory types 38

3.4.2 Whirl trajectories 40

3.4.3 Alternating sign trajectories 42

3.4.4 Mixed trajectories 45

3.5 Outline of an algorithm for finding optimal trajectories for kinematic planar rigid-body systems 47

3.6 Finding best two-stage whirl trajectories 49

3.7 Properties of control line trajectories 51

3.8 Finding best TGT trajectories 53

3.9 Finding best singular trajectories 55

3.10 Finding best regular trajectories 56

3.10.1 Partitioning the Hamiltonian values 57

3.10.2 Reduction to a Lipschitzian optimization problem 58

3.10.3 Finding optimal trajectories for a fixed control line L 60

3.11 Further properties of control line trajectories 62

3.11.1 Notation 63

3.11.2 Parametrization of the control lines 64

3.11.3 Durations of controls 67

3.11.4 Durations for a sequence of controls 69

3.11.5 Finiteness of trajectories corresponding to a control line L . . 70

3.12 Lipschitz continuity 70

3.12.1 Lipschitz continuity of $d(L(H), \mathbf{u}, \mathbf{t})$ and $c(L(H), \mathbf{u}, \mathbf{t})$ 71

3.12.2 Analysis of $t_i(H)$ and $d_i(H)$, $1 < i < n$ 72

3.12.3 Analysis of k_x, k_y, k_θ, and $\text{atan}(k_y, k_x)$ 77

3.12.4 Analysis of q^L 79

3.12.5 Analysis of d_1, t_1, d_n, and t_n 81

3.13 Implementation 86

3.14 Conclusion and future work 87

4 Multi-query for near-optimal trajectories 89

4.1 Introduction 89

4.1.1 Related work 92

4.2 Path convexity and subconvexity 93

4.3 Computing reachable balls 97

4.3.1 Example: 2R planar arm 98

4.3.2 Reachable balls for systems with Lipschitz continuity 100

4.4 Safe covers of free configuration space 100

4.4.1 Reachable hypercubes 101

4.4.2 Uniform grid cover 102

4.4.3 Cube subdivision cover 103

4.4.4 Covering collision space 105

4.4.5 Larger cells, and cell merging 106
4.5 Existence of optimal paths in the cover 107
4.6 Case study: planar arms 109
4.7 Case study: Reeds-Shepp car 110
4.8 Conclusions and future work 112

5 Maximum k-survivability 114

5.1 Introduction 114
5.1.1 Model 116
5.2 Related work 119
5.3 Computing k-survivability 124
5.3.1 Computing k-survivability under the fixed trap model 124
5.3.2 Computing k-survivability under the variable trap model . . . 125
5.4 Theoretical results 125
5.4.1 NP-hardness of kSP 126
5.4.2 Properties of kSP 129
5.5 Algorithms 129
5.5.1 Complete algorithm 130
5.5.2 Heuristic algorithm 133
5.6 Experimental results 137
5.6.1 Experiment setup 137
5.6.2 Results 138
5.7 Conclusion and future work 140
5.7.1 A heuristic for escaping from local maxima for $n > k > 1$ . . . 141

List of Tables

4.1 Performance for robot arm system. 110

4.2 Performance for Reeds-Shepp car. 112

5.1 Running times measured in milliseconds for $k = 1$. 139

5.2 Running times measured in milliseconds for $k = 2$. 139

5.3 Running times measured in milliseconds for $k = 3$. 140

5.4 Running times measured in milliseconds for $k = 4$. 140

5.5 Running times measured in milliseconds for $n = 4$ and $k = 2$. 142

5.6 Running times measured in milliseconds for $n = 5$ and $k = 2$. 142

List of Figures

2.1 Illustration of cell-decomposition method. 8

2.2 Illustration of probabilistic roadmap method. 9

3.1 An approximately optimal trajectories and a chattering trajectory. . . 15

3.2 Approximately optimal trajectories for an omni-directional vehicle. . 30

3.3 Illustration of proof of theorem 3.2. 33

3.4 Illustration of proof of Theorem 3.5. 37

3.5 Taxonomy of optimal trajectories. 38

3.6 An optimal trajectory for the bench mover's problem. 39

3.7 Whirl trajectory for the bench mover's problem. 41

3.8 Distance constraints for whirl trajectories. 42

3.9 A mixed trajectory for bench mover's problem. 46

3.10 An approximately optimal trajectory for a refrigerator. 62

3.11 Illustration of proof of Theorem 3.6 in the case that $g \neq 0$. 64

3.12 Illustration of proof of Theorem 3.11 in the case that u_i is a translation. 73

3.13 Illustration of proof of Theorem 3.11 in the case that u_i is a rotation, $\omega_{i-1} > \omega_i$, and $\omega_{i+1} > \omega_i$. 75

3.14 Illustration of proof of Theorem 3.15 in the case that u_i is a translation. 82

3.15 Illustration of proof of Theorem 3.15 in the case that u_i is a rotation. 83

4.1 A planar arm problem. 90

4.2 Decomposition of the configuration space. 91

4.3 Subconvexity of set X with respect to set Y, under an optimal steering method. 95

4.4 Different size cells at various configurations in the configuration space of the 2R arm. 99

4.5 Illustration of proof of Theorem 4.2. 103

4.6 Example trajectories of 3R and 4R planar arms. 109

4.7 Example trajectories in different environments of the Reeds-Shepp car. 111

5.1 Toy example of a high 1-survivability set of three paths. 115

5.2 Eight paths with high 1-survivability. 117

5.3 Example of high 1-survivability paths in different environments and parameters. 119

5.4 Illustration of maximum diversity problems. 120

5.5 Gadget used in the proof of Theorem 5.1. 127

5.6 Example paths with high k-survivability for different values of k. . . . 130

5.7 All methods' k-survivability at the end of path selection phase, path improvement phase and path shortening phase. 138

Chapter 1

Introduction

One goal of robotics research is to build intelligent robots that are able to accomplish tasks described at a high level while subject to some motion constraints. For example, for a mobile robot, one common task is to move from the current location to another location without colliding with obstacles. After receiving the order, we expect the mobile robot to autonomously send a sequence of commands to its wheels so that the robot reaches the desired location after executing these commands.

Typically, there are many ways for a robot to accomplish one task. When multiple choices are available, picking a "good" solution with respect to some objectives is desirable. For example, in order to save energy, mobile robots may prefer to follow shorter paths, assuming that the path length is proportional to the fuel consumption. Another example is that in order to avoid excessively wearing out the mechanical system, robots may prefer shorter sequences of commands, assuming that the length of the command sequence is proportional to the damage to the mechanical system.

Although defining an objective for the robot is easy in the above examples, it is sometimes unclear how to define a good objective for the robots. For example, for a set of robots moving in a threat environment, is it better for the robots to travel together, or should the robots split up? Within a dangerous environment, how can we define a meaningful objective for a set of robots?

In this thesis, we focus on the problem of finding a feasible path or finding a set of feasible paths while optimizing some objective functions.

We first consider the problem of finding time-optimal trajectories. The time-optimal trajectory problem has been studied extensively, and several general algorithms for finding optimal trajectories with a provable guarantee do exist (Karaman and Frazzoli 2011; Arslan and Tsiotras 2013; Salzman and Halperin 2014; Janson et al. 2015; Otte and Frazzoli 2016). However, these algorithms typically are randomized and are only guaranteed to find an optimal solution as the running time approaches infinity.

On the other hand, for some simple systems, time-optimal trajectories can be found in constant time deterministically based on the properties of the systems (Dubins 1957; Reeds and Shepp 1990; Balkcom and Mason 2002b; Balkcom, Kavathekar, and Mason 2006). This hints that finding an optimal trajectory may be easier when the properties of systems are known. How can we exploit the knowledge about systems to help us design an efficient algorithm for finding optimal trajectories?

We are also interested in multi-robot motion planning problems in a threat environment. When a set of robots are delivering supplies in a threat environment, we can send out more robots than necessary to ensure that a certain amount of supplies is delivered. How can we find paths to ensure that enough robots survive? We consider a novel multi-robot motion planning problem in which the goal is to ensure that a certain number of robots reach the goal.

Researchers have tried to solve problems of finding one safe trajectory for a single robot (Zabarankin, Uryasev, and Murphey 2006; Miller et al. 2011; Boidot, Marzuoli, and Feron 2015; Babel and Zimmermann 2015). However, finding paths for multiple robots in a threat environment is more complicated, since the potential threats may be correlated in some manner. Since the threats may be correlated, what are the safest trajectories for these robots? Is it better for the robots to travel together, or should the robots split up?

Here is a brief summary of the results in the thesis.

Single query for optimal trajectories. We consider the problem of finding optimal trajectories for kinematic planar robots in an unobstructed environment. Although kinematic planar robots are simple, and properties of optimal trajectories are known (Furtuna and Balkcom 2010; Furtuna et al. 2011; Furtuna 2011), no efficient algorithm for finding optimal trajectories exists for kinematic planar robots. Moreover, for some kinematic planar robots and some configurations, optimal trajectories may not exist.

We use a model, the *costly-switch model*, in which a switching cost is charged for each switch of controls. Under the costly-switch model, the existence of optimal trajectories is guaranteed. We characterize necessary conditions for optimal trajectories with switching costs for all kinematic planar robots. Finally, we design an efficient algorithm to find approximately optimal trajectories for all kinematic planar robots within arbitrarily small error tolerance under this model. The details are described in Chapter 4, and this work has been published in ICRA 2014, WAFR 2014, and IJRR (Furtuna et al. 2013; Lyu et al. 2014; Lyu and Balkcom 2014; Lyu and Balkcom 2016).

Multi-query for near-optimal trajectories. For many applications, such as computer games (Botea et al. 2013) and route planning (Delling et al. 2009), users want to find trajectories in an obstructed environment repeatedly, Since the environment will not change over time, it is sometimes beneficial to preprocess the environment and build data structures to represent the configuration space first so that optimal trajectories between any two configurations can be found efficiently. Thus, we consider the problem of finding near-optimal trajectories between different pairs of configurations within the same obstructed environment.

Most known algorithms for finding near-optimal trajectories are sampling-based, and these algorithms are guaranteed to find a near-optimal trajectory as the running time approaches infinity. Since executing a program forever is impractical, we would like to design a deterministic algorithm with a provable guarantee for finding near-optimal trajectories.

Moreover, most known algorithms for finding near-optimal trajectories are very general without using any special property of the systems. However, in order to design an efficient algorithm, exploiting special properties of the system is necessary. For example, for some systems, finding an optimal trajectory connecting two configurations in an unobstructed environment can be solved efficiently (Dubins 1957; Reeds and Shepp 1990; Balkcom and Mason 2002b; Balkcom, Kavathekar, and Mason 2006). For a system with an *optimal steering method*, an algorithm that finds an optimal trajectory connecting two configurations in an unobstructed environment, since the special properties of the system is exploited in the optimal steering method, we would like to treat the optimal steering methods as a black box, and design a generic algorithm for finding near-optimal trajectories assuming an optimal steering method for the system is available.

Based on the idea of the cell-decomposition method, we design an algorithm for decomposing the free configuration space into cells. Within each cell, an optimal steering method can be used to find a collision-free trajectory connecting two configurations in the same cell. We store all cells in a data structure and search this data structure for good paths. The details are described in Chapter 4, and this work has been published in IROS 2015 (Balkcom et al. 2015).

Maximum k-survivability. We consider a new motion planning problem: given two points in a 2D environment with a distribution of obstacles, find n trajectories connecting two given points while maximizing the probability that at least k trajectories reach the goal. Intuitively, we not only need a set of short trajectories so that robots can quickly pass through the dangerous environment, but also a set of diverse trajectories so that trajectories will not be blocked simultaneously by the same obstacle.

This problem is partially motivated by the work of Erickson and LaValle (Erickson and LaValle 2009), in which they proposed the concept of *survivability*. The survivability of a set of trajectories can be considered as the correlation of damage caused by the same obstacle. Moreover, the survivability can be considered as a measure of the diversity for a set of trajectories, since if trajectories are likely to be blocked by the same obstacle, these trajectories have low diversity intuitively. However, survivability does not capture the intuition that longer trajectories are vulnerable, and this motivates us to propose a new definition.

We define *k-survivability* to be the probability that at least k trajectories reach the goal for n robots moving along n trajectories in a threat environment. We formalize the maximum k-survivability problem, and show that maximum k-survivability problem is not only NP-hard but also hard to find approximately optimal solutions. We design a set of heuristic algorithms to find good solutions for the maximum k-survivability problem. The details are described in Chapter 5, and this work has been accepted by RAL (Lyu, Chen, and Balkcom 2016a; Lyu, Chen, and Balkcom 2016b).

Chapter 2

Related work

In this chapter, we briefly summarize related work in motion planning.

2.1 Complete motion planning

Motion planning problems usually have the following form: given a system and an environment, find a valid trajectory connecting two given configurations. A motion planning algorithm is called *complete* if the algorithm is guaranteed to terminate within a finite time and find a valid trajectory whenever a trajectory exists. Although the motion planning problem was proved to be PSPACE-hard in general, several complete motion planning algorithms have been developed (Collins 1975; Schwartz and Sharir 1983; Ó'Dúnlaing 1987; Canny 1988; Canny, Rege, and Reif 1991; Chazelle et al. 1991; Canny 1993; Halperin and Sharir 1996; Basu, Pollack, and Roy 2000; Trinkle and Milgram 2002; Varadhan et al. 2006; Basu, Pollack, and Roy 2006; Shvalb et al. 2007; Basu et al. 2014; Basu and Roy 2014). Since these method are based on analytical representation of the configuration space, due to the curse of dimensionality,

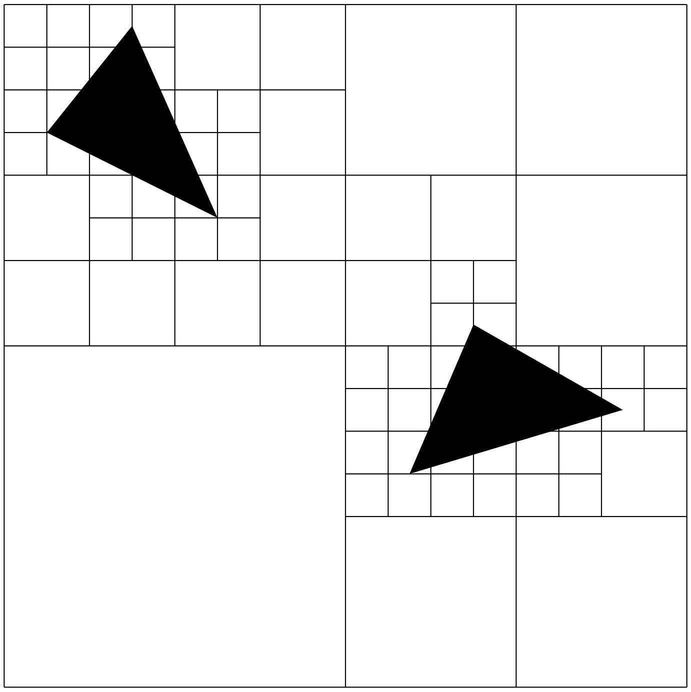


Figure 2.1: Illustration of the cell-decomposition method. Black objects are obstacles. The configuration space is divided into several cells with varied sizes. Each cell is either free or obstructed.

these methods can only solve problems in a low dimensional configuration space within reasonable time. LaValle gives a comprehensive survey of motion planning algorithms (LaValle 2006).

2.2 Resolution complete motion planning

Since finding a complete algorithm is inherently difficult, researchers have tried to relax the definition of completeness. A motion planning algorithm is called *resolution complete* if the algorithm is guaranteed to terminate within a finite time and finds a valid trajectory when the resolution is fine enough. Many resolution complete algorithms are based on *cell-decomposition* method: partition the configuration space

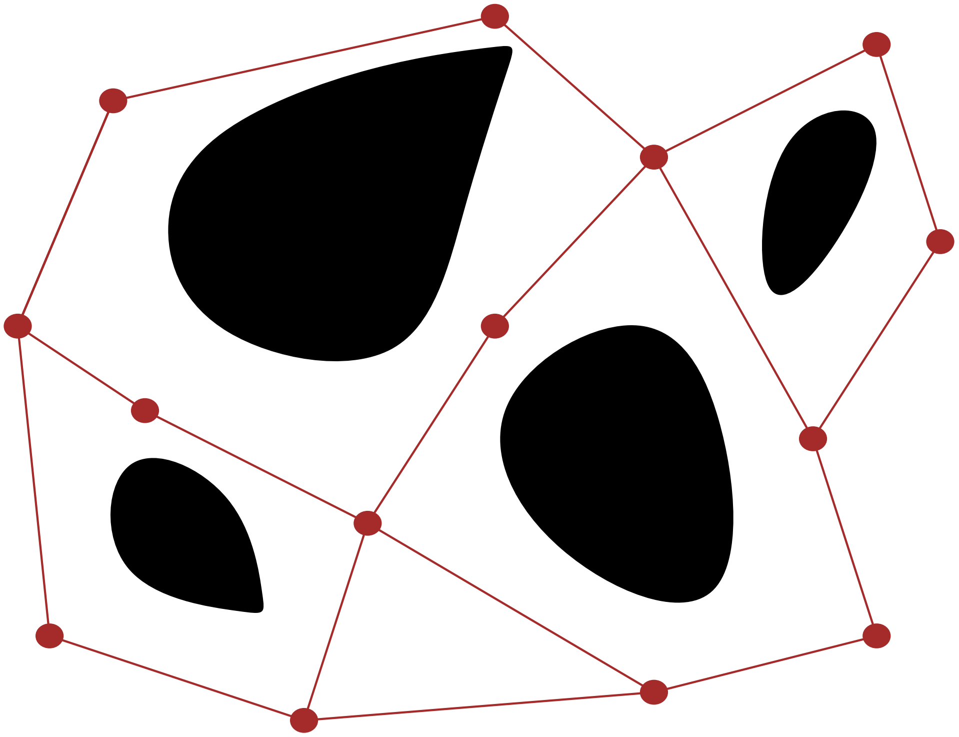


Figure 2.2: Illustration of the probabilistic roadmap method. Random samples are generated in the configuration space and connections between samples are created.

into several cells and search for a path among them; see Figure 2.1. Yershov and LaValle give sufficient conditions for the existence of resolution complete algorithms (Yershov and LaValle 2010). Although cell-decomposition method works well for some problems (Brooks and Lozano-Pérez 1985; Zhu and Latombe 1991; Barraquand, Langlois, and Latombe 1992; Chen and Hwang 1998; Porta et al. 2007; Zhang, Kim, and Manocha 2007), cell-decomposition algorithms are still suffer from the curse of dimensionality.

2.3 Probabilistically complete motion planning

Another way to relax the definition of completeness is to exploit the power of randomness. A randomized motion planning algorithm is called *probabilistically complete* if the algorithm is guaranteed to find a valid trajectory with probability one as the running time approaches infinity, if a valid trajectory exists. There are two major classes of randomized motion planning algorithms: probabilistic roadmaps (PRM, Kavraki et al. 1996) and rapidly exploring random trees (RRT, LaValle and J. Kuffner Jr. 2001); both are probabilistically complete. The idea of the probabilistic roadmap method is to generate a set of random samples and build a graph structure over them; see Figure 2.2. Then, a trajectory between two configurations can be found by searching the graph. There are many studies and extensions for PRM and RRT (Barraquand et al. 1997; Svestka and Overmars 1997; Hsu, Latombe, and Motwani 1999; Siméon, Laumond, and Nissoux 2000; Hsu et al. 2002; Hsu, Latombe, and Kurniawati 2006). Combining sampling-based algorithms with cell-decomposition approaches has also been explored (Lingelbach 2004; Rosell and Iñiguez 2005).

2.3.1 Asymptotically optimal motion planning

In practice, users may not be satisfied with a valid trajectory, but seek an optimal trajectory connecting configurations. Recently, Karaman and Frazzoli proposed two sampling-based algorithms that are guaranteed to find an approximately optimal trajectory as the running time approaches infinity (Karaman and Frazzoli 2011). There are several variants of Karaman and Frazzoli's algorithm (Arslan and Tsiotras 2013; Salzman and Halperin 2014; Janson et al. 2015; Otte and Frazzoli 2016).

2.4 Topological analysis in motion planning

Analyzing topological properties of configuration space plays an important role in robotics research (Farber 2003; Farber and Grant 2008; Farber 2008; Pokorny, Hawasly, and Ramamoorthy 2016), and several motion planning algorithms are based on the topological properties of configuration spaces (Bhattacharya, Likhachev, and Kumar 2012; McCarthy, Bretl, and Hutchinson 2012; Kim et al. 2012; Bhattacharya et al. 2013; Kim et al. 2013; Kim, Bhattacharya, and Kumar 2014; Kuderer et al. 2014; Hernández, Carreras, and Ridao 2015). Since the number of distinct homotopy classes of trajectories can be used as a definition of the diversity of trajectories, Schmitzberger *et al.* studied the homotopy classes of roadmaps (Schmitzberger et al. 2002). However, Nieuwenhuisen and Overmars pointed out that since two homotopic trajectories in higher dimension may be hard to deform, it may be too coarse to classify trajectories by homotopy classes in motion planning (Nieuwenhuisen and Overmars 2004). Thus, instead of using homotopy classes, Jaillet and Simeon proposed the concept of *first-order deformation* to classify trajectories (Jaillet and Siméon 2008).

2.5 Nonholonomic motion planning

A system is called *nonholonomic* if the differential constraints of the system cannot be described by configuration space variables (Bloch et al. 2007). Nonholonomic systems have been studied from the differential geometry perspective and from the control theory perspectives.

2.5.1 Differential geometry

Due to nonholonomic constraints, a trajectory connecting initial and goal configurations may not exist even when there are no obstacles. Fortunately, for some systems, the existence of trajectory from any initial configuration to all connected configurations can be proven by the Chow-Rashevsky theorem (Calin and Chang 2009). A connected configuration space with the metric induced by shortest trajectories forms a *sub-Riemannian manifold*, and geometric properties have been studied (Bellaïche and Risler 1996; Montgomery 2002). Jean gives a comprehensive discussion of the relationship between sub-Riemannian geometry and motion planning (Jean 2014).

2.5.2 Control theory

Nonholonomic motion planning has also been studied in control theory and robotics (Lafferriere and Sussmann 1991; Lafferriere and Sussmann 1993; Li and Canny 1993; Barraquand and Latombe 1993; Murray and Sastry 1993; Laumond et al. 1994; Bushnell, Tilbury, and Sastry 1995; Laumond and Risler 1996; Zefran and Kumar 1997; Sekhavat et al. 1998). Due to nonholonomic constraints, optimal trajectories connecting two given configurations may not exist, even when there are no obstacles. Fortunately, for some systems, the existence of optimal trajectories can be proven by the Filippov theorem (Laumond 1998).

In cases where the Filippov theorem does not apply, there exist no optimal trajectories, since for any trajectory with finite number of switches, there exists a faster trajectory with more number of switches; this phenomenon is called *chattering* in control theory. When chattering occurs, although the infimum of the cost of all feasible trajectories exist, there is no trajectory can achieve the infimum; but the infimum can be approximated arbitrarily by a trajectory with finite number of switches under some assumptions (Berkovitz and Medhin 2012). Zelikin and Borisov give a comprehensive discussion about the chattering phenomenon in control theory (Zelikin and Borisov 1994).

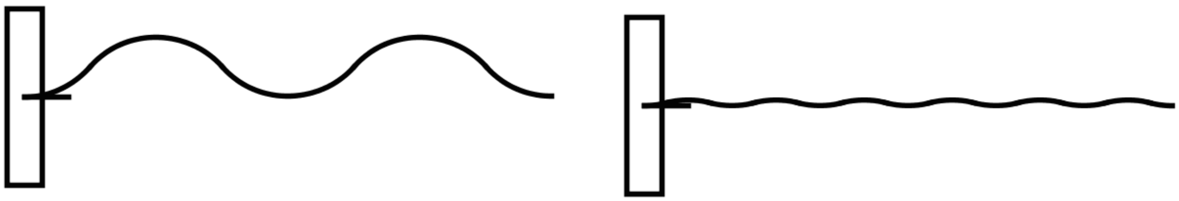
Chapter 3

Single query for optimal trajectories

3.1 Introduction

In order to reach the goal efficiently, one natural objective for motion planning is to find a time-optimal trajectory between two given configurations. However, for some systems and some configurations, an optimal trajectory may not exist. The non-existence of solutions becomes one difficulty of finding optimal trajectories. In this chapter, we provide a model in which the existence of optimal trajectory is guaranteed and an algorithm to find approximately optimal trajectories under this model.

Consider a toy problem for demonstration the non-existence of optimal trajectories. One mover wants to move a bench (modeled as a line segment) from one location and orientation to another, as efficiently as possible. Since the bench is too heavy, the bench can only be moved by lifting one end and rotating the bench around the end that is still on the ground, with rotational velocity of ±1. The mover wishes to find a sequence of durations and directions of rotations that brings the bench to the final configuration, while minimizing the sum of the absolute values of the angles



(a) Costly-switch model with switching cost 1.

(b) Chattering.

Figure 3.1: Trajectories for a bench starting at (-4, 0, 0). When the switch costing is one, the optimal trajectory takes 5 actions; see Figure 3.1a. However, if there is no switching cost, by increasing number of actions, it is always possible to create a faster trajectory; see Figure 3.1b.

rotated through. This problem is related to the Reeds-Shepp problem (Reeds and Shepp 1990) of finding a shortest path for a steered car, but with only four discrete controls, none of which is a pure translation.

No optimal trajectory with a finite number of actions exists when the mover wants to move the bench in a straight line: for any trajectory with finitely many switches, a faster trajectory with more switches always exists, a phenomenon called *chattering*. When chattering occurs, the bench mover is required to run back and forth between ends of the bench infinitely many times, rotating the bench by an infinitely small angle; see Figure 3.1.

The chattering phenomenon is a fundamental problem in robot motion planning. Sussmann shows that an extension of the well-known Dubins car (Dubins 1957) to include bounds on angular acceleration leads to chattering (Sussmann 1997). Moreover, Desaulniers shows that chattering may occur if there are obstacles in the environment, even for Reeds-Shepp car (Reeds and Shepp 1990) that are well-behaved without obstacles (Desaulniers 1996).

Since optimal trajectories may not exist, no algorithm can be guaranteed to find optimal solutions. One remedy is to require motion planners to find a “good enough” trajectory when no optimal solution exists. However, systems that chatter also tend to expose weaknesses in the model that may not have been immediately apparent. A trajectory with very many turns is in fact quite bad for the bench mover, even if it is “short" in configuration space.

A natural, although certainly imperfect, approach to avoid having a very large number of discontinuous switches between discrete controls is to charge a fixed cost for switches; this approach has been used in practice at least as far back as (Barraquand and Latombe 1991; Stewart 1992). Choosing a finite set of primitive controls is a required first step for many general-purpose approaches to non-holonomic motion planning, including, for example, RRT-type motion planners (LaValle and J. Kuffner Jr. 2001). Moreover, picking a set of primitives not only makes it easier to model the switching cost, but also makes it easier to explain the system and implement trajectories, since a trajectory can be represented by a sequence of controls and a sequence of durations.

We limit the choice of controls to certain motion primitives. With this set of primitives, we associate each pair of switch of controls with a predetermined fixed cost, which is a positive real number. This predetermined cost may be suggested naturally by the design of the robot (for example, time cost of running between ends of a bench), or may be selected more arbitrarily to indicate a user preference for trajectories with fewer switches. This fixed cost both avoids chattering, and penalizes otherwise un-modeled costs like the cost of wearing out a switching mechanism.

In order to make the consideration of switching costs more concrete, we focus on finding time-optimal trajectories for kinematic planar rigid bodies. Rigid bodies are building blocks for many models of robotic locomotion or manipulation systems, and the time-optimal trajectories for the case of zero switching costs have been already studied (Furtuna 2011).

Given a set of primitives, a trajectory is defined by a start configuration, and a sequence of motion primitives, each executed for some particular duration. We can view the optimal control problem as having two parts: selecting a sequence of primitives (the *discrete structure* of the trajectory), and choosing each duration (from a continuous interval).

Solution approach. We first characterized a set of necessary conditions of optimal trajectories for kinematic planar rigid bodies under the costly-switch model by using *Blatt's Indifference Principle* (BIP, Blatt 1976). In comparison to our necessary conditions with the necessary conditions of optimal trajectories for kinematic planar rigid bodies under the cost-free-switch model (Furtuna 2011) derived by using Pontryagin's Maximum Principle (PMP, Pontryagin et al. 1962), the necessary conditions derived by using BIP are weaker than those given by PMP. Although the necessary conditions tell us much about the continuous durations along trajectories, they do not constrain the discrete structures as strongly.

After deriving necessary conditions using BIP, we work through an example of applying necessary conditions directly to analyze time-optimal trajectories for the relatively simple bench-mover's problem described above. However, due to the lack of constraints on trajectory structures, it appears very difficult to find similarly strong analytical results for more complicated systems, including other kinematic planar rigid bodies.

To attack the problem of finding optimal trajectories for kinematic planar rigid bodies with a specified set of primitives, we use BIP to classify trajectories into several types, which will be described later. For all but one of these classes, durations can be computed exactly, and we can use an A* search to search over trajectory structures. For the last remaining class, we show that Lipschitzian optimization techniques can be used to find provably good numerical approximations for the durations, while applying a different A* search over trajectory structures.

Limitations. Although we believe that this work represents an interesting new exploration the connection between motion planning and optimal control, we admit that the algorithmic techniques presented here suffer from some limitations. Many of these limitations do suggest rich problems for future study.

The focus on kinematic planar rigid bodies with a time metric is limiting. Extending work on cost-free-switch models beyond simple systems or simple metrics using PMP has proved challenging, because optimal trajectories for more complex systems may not be described analytically in an easy way. However, choosing discrete (perhaps piecewise-constant) controls with a cost of switching ensures that optimal trajectories are describable by recognizable functions, and we believe that the current techniques (particularly including application of Karush-Kuhn-Tucker conditions, which do not require integration of an adjoint vector) could be extended to more interesting systems.

The unobstructed environment is limiting, but we do not believe this is a fundamental limitation. Although optimal trajectories may not exist in an obstructed environment under the cost-free-switch model, optimal trajectories certainly exist for strictly positive switching costs even in the presence of obstacles (Furtuna 2011). Moreover, we believe that the result of BIP could be extended to allow for state constraints.

Perhaps the main limitation of the algorithms in this work is computational cost. Due to the relative weakness of BIP w.r.t. PMP, the length of sequences of primitives generated in the algorithm may be exponential in the number of primitives, while they are only polynomial under the cost-free-switch model. Furthermore, the length of sequences of primitives generated in the algorithm increases when the cost of switch decreases. Hence, finding approximately optimal trajectories with many primitives is computationally infeasible by this method, and we only use simple systems to demonstrate our technique. We believe that good heuristics for the A* search over discrete trajectory structures may ameliorate this issue.

3.1.1 Model and notation

We use q to denote a configuration of the system and use u to denote a control in the control space U, which contains finite number of primitives.

At a configuration q, if we apply a control u, the instantaneous configuration space velocity, $\dot{q}$, can be expressed as a function f, such that $\dot{q} = f(q, u)$. A trajectory can be represented as a pair of sequences $(\mathbf{u}, \mathbf{t})$ with the start configuration q_s, where $\mathbf{u} \in U^n$ is a sequence of controls, $\mathbf{t} \in \mathcal{R}_+^n$ is a sequence of durations, and n is the length of the sequence. When the start configuration is clear from the text, we use $(\mathbf{u}, \mathbf{t})$ to denote a trajectory.

We model the cost of switching between controls as a function $C : U \times U \to \mathcal{R}_+$ that depends on the control applied before and the control applied after. Furthermore, we assume that for any three controls u_a, u_b, and u_c, the cost of switching between controls satisfies the *triangle inequality*, $C(u_a, u_b) + C(u_b, +u_c) \geq C(u_a, u_c)$, to ensure that switching from u_a to u_c directly is always faster than switching to u_c through other intermediate controls. The cost of a trajectory is the summation of all durations and all switching costs of the trajectory.

Problem statement: given a start configuration q_s, a final configuration q_f, a finite control set U, and a cost function C, find a trajectory $(\mathbf{u}, \mathbf{t})$ with minimum cost, connecting q_s to q_f.

3.1.2 Related work

We briefly summarize related work in optimal control.

Optimal control

Finding optimal trajectories connecting start and goal configurations has been studied in optimal control theory (Schättler and Ledzewicz 2012). There are two major methods to obtain analytical solutions: the direct method and the indirect method. Since finding optimal trajectories analytically is extremely difficult, only a few systems' optimal trajectories are fully characterized. In practice, we would like to develop an efficient numerical method to find approximately optimal trajectories with provable guarantees. In this section, we survey results in both optimal control theory and numerical optimal control theory.

Direct. The direct method is to use *dynamic programming* (Bellman 2010) techniques to set up a partial differential equation, *Hamilton-Jacobi-Bellman* (HJB) equation, whose solution is an optimal trajectory. However, classical solutions for the HJB equation usually do not exist. Thus, Lions and Crandall introduced the concept of a *viscosity solution*, which is a generalization of the classical solution (Evans 2010).

Nonetheless, for many systems, analytical solutions are still hard to obtain, even though viscosity solutions exist (Bardi and Capuzzo-Dolcetta 1997).

Indirect. The indirect method is to characterize necessary conditions for optimal trajectories first and then find the best trajectory among all trajectories that satisfy necessary conditions. First-order necessary conditions are usually characterized by using Pontryagin's Maximum Principle (PMP, Pontryagin et al. 1962), one of the most powerful tools in optimal control theory. Higher-order necessary conditions for optimal trajectories have been studied as well (Krener 1977; Osmolovskii and Maurer 2012). PMP can also deal with optimal control problems with state constraints (Hartl, Sethi, and Vickson 1995; Milyutin and Osmolovskii 1998), problems in infinite-dimensional spaces (Krastanov, Ribarska, and Tsachev 2011), or optimal control problems for hybrid systems (Sussmann 2000; Garavello and Piccoli 2005; Dmitruk and Kaganovich 2008). Liberzon gives one simplified proof of PMP (Liberzon 2012). Although PMP provides fruitful information about optimal trajectories and many simple systems' optimal trajectories are analyzed in this manner, obtaining analytical solution for complex systems is still intractable. Clarke gives a comprehensive discussion about the relationship between PMP and dynamic programming (Clarke 2013).

Kinematic planar robot. For some kinematic planar robots, optimal trajectories can be found analytically. Usually, researchers characterized optimal trajectories for a specific system in the following way. First, a sufficient family of optimal trajectories is characterized. When a necessary family of optimal trajectories is found, an analytical solver could enumerate all trajectories in this family and then determine an optimal trajectory among them. The efficiency of the analytical solver depends on the size of the family, and the size of the family may be large, since the family may not be necessary.

Second, a sufficient and necessary family of optimal trajectories is characterized and this directly leads to a better analytical solver. Finally, assuming the goal is in the origin, the configuration space can be partitioned into parts such that the all optimal trajectories for all configurations in one part have an identical first control with possibly different durations. This partition of the configuration space is called the *optimal trajectory synthesis* of the system. Based on the optimal trajectory synthesis, an efficient and analytical solver can possibly be developed.

We briefly summarize known results for kinematic planar robots.

(a) Dubins car: Dubins firstly characterized optimal trajectories for a car that can only drive forward (Dubins 1957). The optimal trajectory synthesis for Dubins car has also been fully studied (Bui et al. 1994). Based on this optimal trajectory synthesis, a real-time optimal trajectories solver has developed (Shkel and Lumelsky 2001). The problem of Dubins car with acceleration has also been studied (Sussmann 1997). Finding optimal trajectories for Dubins car among obstacles has been shown to be NP-hard (Reif and Wang 1998) and several algorithms has been proposed (Agarwal and Wang 2001; Agarwal et al. 2002; Boissonnat and Lazard 2003; Ny, Feron, and Frazzoli 2012; Goaoc, Kim, and Lazard 2013).

(b) Reeds-Shepp car: Reeds and Shepp characterized a sufficient family of optimal trajectories for a car that can drive forward and backward (Reeds and Shepp 1990). A sufficient and necessary family of optimal trajectories has been found

(Sussmann and Tang 1991; Boissonnat, Cérézo, and Leblond 1994), and the optimal trajectories synthesis of Reeds-Shepp car has also been determined (Desaulniers and Soumis 1995; Souères and Laumond 1996; Souères and Boissonnat 1998; Souères 2007). Based on this optimal trajectory synthesis, an efficient optimal trajectories solver has been developed (Wang, Chen, and Souères 2009). Finding optimal trajectories for Reeds-Shepp car among obstacles has also been studied (Desaulniers 1996; Desaulniers, Soumis, and Laurent 1998).

(c) differential-drive: a sufficient and necessary family of optimal trajectories has been characterized (Balkcom and Mason 2000a). The optimal trajectories synthesis has been characterized as well (Balkcom and Mason 2000c; Balkcom and Mason 2000b; Balkcom and Mason 2002b), which directly leads to an efficient analytical solver.

(d) omni-directional vehicle: a sufficient family of optimal trajectories has been characterized (Balkcom, Kavathekar, and Mason 2006), and an analytical solver has been developed (Wang and Balkcom 2012a).

We and many other researchers have tried to generalize techniques, typically based on Pontryagin's Maximum Principle (Pontryagin et al. 1962), aiming to gain a greater understanding of optimal motion for mobile robots (Cockayne and Hall 1975; Reister and Pin 1994; Moutarlier, Mirtich, and Canny 1996; Renaud and Fourquet 1997; Chyba and Sekhavat 1999; Vendittelli, Laumond, and Nissoux 1999; Balkcom and Mason 2002a; Chyba and Haberkorn 2006; Giordano et al. 2006; Chitsaz et al. 2009; Chitsaz 2008; Giordano and Vendittelli 2009; Furtuna and Balkcom 2010; Furtuna et al. 2011; Furtuna 2011; Wang and Balkcom 2012a; Wang and Balkcom 2012b; Lyu et al. 2014; Lyu and Balkcom 2014; Lyu and Balkcom 2016). Boscain and Piccoli gives more examples for control systems on 2-d manifolds (Boscain and Piccoli 2004).

Numerical optimal control

Since obtaining optimal trajectories analytically is difficult, many numerical methods of finding optimal trajectories have been proposed. Here, we are more interested in complete and exact numerical solvers, that is, solvers that find an approximately optimal trajectory within a given error tolerance in finite time assuming no floating-point error. Based on the underlying principle used in the numerical solvers, numerical optimal control solvers can be categorized into three classes: direct, indirect, and spectral.

Direct. The direct solvers use *discretize-then-optimize* approach that discretizes the optimal control problem in a sufficient resolution and then find an optimal solution for the discretized problem (LaValle and Konkimalla 2001; Yershov and Frazzoli 2016). For example, optimal trajectories for Dubins car and Reeds-Shepp can be obtained by solving corresponding HJB equations numerically (Takei and Tsai 2013). The result of the direct approach is usually a *optimal control policy* that maps from free configurations to a control that should be applied in order to reach the goal optimally. Hence, the direct approach may spend much time computing the policy for configurations that are unnecessary for the given start configuration.

Indirect. On the other hand, the indirect solvers use *optimize-then-discretize* approach that discretizes the optimal control problem subject to necessary conditions in a sufficient resolution and then find an optimal solution for the discretized problem.

For example, we developed a numerical solver for kinematic planar rigid body (Lyu and Balkcom 2016) based on necessary conditions for optimal trajectories (Furtuna and Balkcom 2010; Furtuna et al. 2011; Furtuna 2011). In order to design an indirect method, the solver designers must carefully analyze the system to obtain necessary conditions. However, there exist *singular trajectories* about which the indirect methods cannot obtain useful information during the search procedure, and this is the major difficulty for applying indirect method. Bonnard and Chyba give a comprehensive discussion about singular trajectories in control theory (Bonnard and Chyba 2003).

Pseudospectral method. Usually, the direct approach leads to a straightforward formulation, but the solution may not satisfy necessary conditions for optimal trajectories. On the contrary, the indirect approach searches for a solution satisfying necessary conditions, but the necessary conditions are usually hard to deal with and introduce numerical instability issues during the search. Besides direct and indirect approaches, there is one alternative method, the *pseudospectral method*, using the *covector mapping principle*, which is a connection between the direct method and the indirect method. Ross and Karpenko give a good review of pseudospectral method (Ross and Karpenko 2012).

Costly-switch model

The problem of costly switches has been studied in the optimal control community with results dating back as far as the 1970s. However, we are aware of little work in the robotics community providing strong results on optimal trajectories with a cost of switches; a notable exception is Stewart's work that uses a dynamic-programming approach to find optimal trajectories under the costly-switch model (Stewart 1992).

One of the most powerful tools for solving optimal control problems, Pontryagin's Maximum Principle (PMP, Pontryagin et al. 1962), does not appear to be the right tool to characterize optimal trajectories under the costly-switch model due to the discontinuity with respect to time in the control function and cost function. Blatt proposed a model in which the control set contains certain primitives (a discrete set of actions), and there is some fixed cost associated with switching between controls (Blatt 1976).

Blatt characterized a set of necessary conditions for optimal trajectories under the costly-switch model; these necessary conditions are known as Blatt's Indifference Principle (BIP). Blatt showed that optimal trajectories always exist and the number of actions must be finite. Blatt's necessary conditions are similar to, but weaker than, those provided by PMP; using BIP to solve an optimal control problem is more challenging than using PMP under the cost-free-switch model. In Blatt's model, the control set is a discrete set, but other models have been proposed as well (Noussair 1977; Kibalczyc and Walczak 1984; Matula 1987). As it turns out, Blatt's Indifference Principle can be considered as a special case of PMP for hybrid systems (Sussmann 2000).

Although the costly-switch model was proposed in the '70s, no algorithms for finding optimal trajectories under the costly-switch model were proposed until the '90s (Teo and Jennings 1991; Stewart 1992); several algorithms have been developed recently (Loxton, Lin, and Teo 2013; Yu et al. 2013). These recent approaches are based on approximating the control function as a piecewise-constant functions, and applying global optimization techniques to find optimal solutions. These algorithms converge to optimal solutions as the number of iterations approaches infinity, but cannot guarantee a bound of error within finite time.

3.2 Mathematical background

In this section, we review two mathematical tools for optimal control and non-linear programming: Blatt's Indifference Principle (Blatt 1976) and Karuhn-Kush-Tucker conditions (Bazaraa, Sherali, and Shetty 2006).

For the rigid-body system studied in this work, the Indifference Principle is sufficient. However, it is interesting that once a particular sequence of discrete, constant controls have been selected for a trajectory, the problem of selecting durations for each control is simply a finite constrained non-linear optimization problem for which KKT may be applied. Although we have used both approaches to derive similar results, the KKT approach is simpler in that it does not require the analytical integration of an adjoint vector – it is for this reason that we present both approaches.

3.2.1 Blatt's indifference principle

Under the costly-switch model, BIP provides a set of necessary conditions for optimal trajectories for any finite dimension configuration space with a finite control set (Blatt 1976). The configuration and control over time of an optimal trajectory $(\mathbf{u}^*, \mathbf{t}^*)$ from a start configuration q_s can be represented as two functions $q^*(t)$ and $u^*(t)$, where $q^*(t)$ and $u^*(t)$ are the configuration of the robot and the control at time t respectively. BIP states that:

(a) There exists a continuous adjoint function $\lambda(t)$, which is non-trivial.

(b) The adjoint function satisfies

$$\frac{d\lambda}{dt} = \frac{\partial}{\partial q} H(\lambda(t), q^*(t), u^*(t))$$

where H is the *Hamiltonian*, which is the product of the velocity in the world frame and the adjoint function.

(c) At the time $\hat{t}$ of switching control u to u', the Hamiltonian is indifferent to both u and u':

$$H(\lambda(\hat{t}), q^*(\hat{t}), u) = H(\lambda(\hat{t}), q^*(\hat{t}), u'), \tag{3.1}$$

and the Hamiltonian function is a positive constant along the trajectory.

Comparing with Pontryagin's maximum principle. Under the cost-free-switch model, PMP (Pontryagin et al. 1962) provides a set of necessary conditions on optimal trajectories, that the first two conditions are the same as BIP, but the third condition is different: the control $u^*(t)$ maximizes the Hamiltonian along the trajectory.

(c) The control $u^*(t)$ maximizes the Hamiltonian along the trajectory:

$$u^*(t) = \arg\max_{u \in U} H(\lambda(t), q^*(t), u), \tag{3.2}$$

and the Hamiltonian function is a positive constant along the trajectory.

That is, PMP requires that the control maximizes Hamiltonian along the trajectory, but BIP only requires that the Hamiltonians of controls are the same at the time of switch. Hence, PMP provides the constraint on the controls of optimal trajectories can use, but BIP only provides the conditions on the duration for a control can apply.

3.2.2 Karush-Kuhn-Tucker conditions

The Karush-Kuhn-Tucker conditions (Bazaraa, Sherali, and Shetty 2006) provide a set of necessary conditions for optimal solutions of constrained non-linear optimization problems. Consider a non-linear optimization problem as follows:

$$\begin{aligned}
\text{minimize} \quad & f(x) \\
\text{subject to} \quad & q(x) = 0 \\
& g(x) \leq 0 \\
& x \in \mathcal{R}^n, \text{with differentiable } f : \mathcal{R}^n \to \mathcal{R}, \\
& \qquad q : \mathcal{R}^n \to \mathcal{R}^m, \text{ and } g : \mathcal{R}^n \to \mathcal{R}^p \ .
\end{aligned}$$

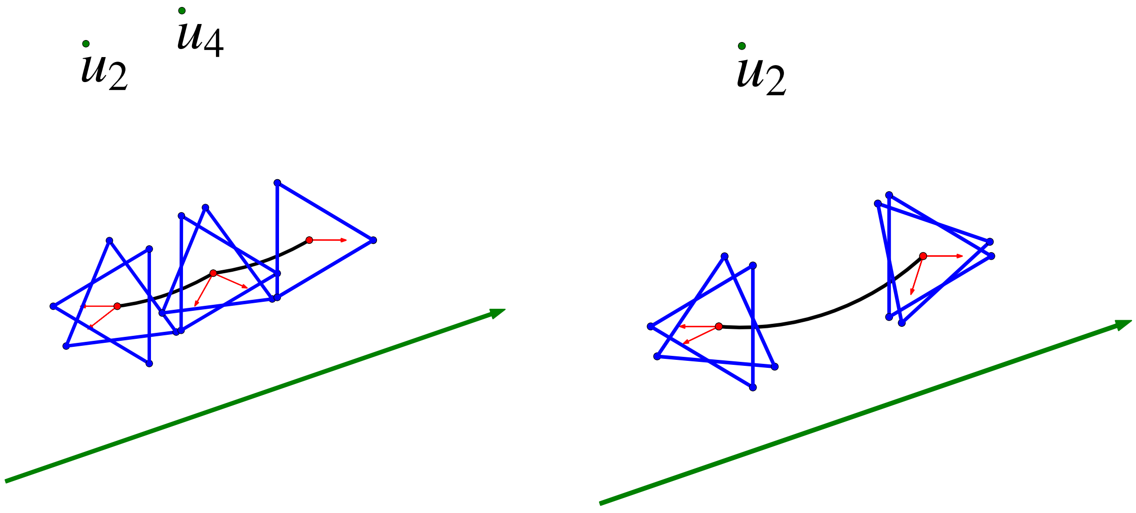
The Karush-Kuhn-Tucker conditions are: If $\hat{x}$ is a local minimum and satisfies *constraint qualification conditions*, then there exists $\lambda \in \mathcal{R}^m$ and $\mu \in \mathcal{R}^p$, such that

(a) $\nabla f(\hat{x}) + \lambda \cdot \nabla q(\hat{x}) + \mu \cdot \nabla g(\hat{x}) = 0.$

(b) $\mu \geq 0.$

(c) $\mu \cdot g(\hat{x}) = 0.$

In order for these conditions to hold, certain *constraint qualification conditions* must be satisfied; since the primary focus of this work is BIP, we omit discussion of constraint qualification.



(a) Trajectory in cost-free-switch model.

(b) Trajectory in costly-switch model.

Figure 3.2: Trajectories for an omni-directional vehicle starting at (-3, -1, π). For the cost-free-switch model, the optimal trajectory takes 5 actions. For the costly-switch model, the (approximately) optimal trajectory takes 3 actions. Thick lines are control lines.

3.3 Necessary conditions for optimal trajectories

In this section, we derive necessary conditions for optimal trajectories for kinematic planar rigid bodies under the costly-switch model. Based on these necessary conditions, we classify optimal trajectories into several classes and we also show that in order to find optimal trajectories, it suffices to find optimal trajectories in some trajectories classes. Since we focus on kinematic planar rigid-body robot, the configuration space is $SE(2)$ and we use $u = (v_x, v_y, \omega) \in \mathcal{R}^3$ to denote a control: x and y velocities in a frame attached to the body (robot frame), and angular velocity. Let U be the control space containing a finite number of primitives: constant-control actions. For example, one action might be $(v_x, v_y, \omega) = (1, 0, 0)$, corresponding to driving in a straight line.

3.3 Necessary conditions for optimal trajectories

Due to the similarity between BIP and PMP, several results under the cost-free-switch model (Furtuna 2011) can be extended to the costly-switch model by similar mechanisms.

Theorem 3.1. *For any kinematic planar rigid body under the costly-switch model, any optimal trajectory* $(\mathbf{u}^*, \mathbf{t}^*)$ *with* n *actions satisfies the following property: there exist four constants* $H > 0$, k_x, k_y, *and* k_θ, *such that for any control* u_i^*, $1 \leq i \leq n$, *with the instantaneous velocity* (v_x, v_y, ω) *in the world frame when* u_i *is applied at a configuration* (x, y, θ), *we have*

$$k_x v_x + k_y v_y + \omega(k_x y - k_y x + k_\theta) = H, \textit{where } k_x^2 + k_y^2 \in \{0, 1\}. \tag{3.3}$$

We also can derive the same result by applying KKT conditions (assuming constraint qualification holds), by fixing the sequence of controls in a trajectory, and showing that because the sequence is arbitrary, the result holds across all trajectory structures. The variables k_x, k_y, and k_θ are Lagrange multipliers from the KKT conditions; in Blatt's indifference principle, they arise as constants of integration.

A trajectory $(\mathbf{u}, \mathbf{t})$ is called *extremal* if there exist four constants $H > 0$, k_x, k_y, and k_θ, such that Equation 3.3 is satisfied.

Equation 3.3 is virtually identical to the necessary conditions derived by using PMP under the cost-free-switch model (Furtuna 2011), except that controls do not necessarily maximize the Hamiltonian. Instead, under the costly-switch model, the Hamiltonian needs only to be a constant throughout the trajectory. Due to this similarity, a geometric structure, similar to the geometric structure under the cost-free-switch model, exists under the costly-switch model.

Geometric structure. If $k_x^2 + k_y^2 = 1$, then the expression $k_x y - k_y x + k_\theta$ in Equation 3.3 can be interpreted as computing the distance of a point (x, y) from some line described by constants k_x, k_y, and k_θ. We therefore call such a trajectory a *control line trajectory*. An extremal trajectory with $k_x^2 + k_y^2 = 0$ is called a *whirl* trajectory because the angular velocity of the rigid body is constant and non-zero over the trajectory.

3.3.1 Control line trajectories

Under the costly-switch model, a nice geometric interpretation for Theorem 3.1 when $k_x^2 + k_y^2 = 1$ is available, similar to the control line interpretation under the cost-free-switch model (Furtuna 2011). For a control line trajectory $(\mathbf{u}, \mathbf{t})$, we define its corresponding *control line*, represented as (k_x, k_y, k_θ) as a directed line in the plane with heading (k_x, k_y) and distance $|k_\theta|$ from the origin. Now, consider Equation 3.3. The term $k_x v_x + k_y v_y$ becomes the translational velocity along the vector (k_x, k_y) and the term $k_x y - k_y x + k_\theta$ becomes the *signed distance* from the reference point of the robot to the control line. By Corollary 1 in Furtuna's Ph.D. thesis (Furtuna 2011), when a rotation is applied, the signed distance from the rotation center to the control line is always H/ω. Similarly, when a translation is applied, the dot product between (k_x, k_y) and (v_x, v_y) must be the Hamiltonian value H. See Figure 3.2 for an (approximately) optimal trajectory for an omni-directional vehicle with control lines under the cost-free-switch model and under the costly-switch model. When the switching cost is introduced, optimal trajectories tend to use fewer number of switches.

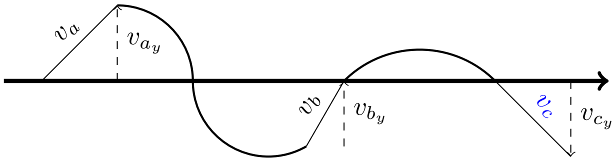


Figure 3.3: Illustration of proof of theorem 3.2: a trajectory containing three actions of translations, v_a, v_b, and v_c. The sign of v_{ay} and v_{by} are the same.

Necessary conditions for control line trajectories

We prove a further necessary condition for a control line trajectory to be optimal.

Theorem 3.2. *For any kinematic planar rigid body under the costly-switch model, any optimal control line trajectory has either zero translation actions, one translation action, or two non-parallel translation actions.*

Proof. Let $g = (\mathbf{u}, \mathbf{t})$ be a control line trajectory. Suppose that g is an optimal trajectory for some initial and goal configurations. We first show that g cannot have more than 1 parallel translation actions. Then, we show that g cannot have more than two non-parallel translation actions.

Suppose that g has two parallel translation actions. Let v_a and v_b be the velocity vectors in the world frame of two non-parallel translation actions of g. We can remove the action of v_b from g and increase the duration of v_a to $t_a + t_b$. The resulting trajectory still reaches the goal but has one fewer control and hence has smaller cost. This contradicts the optimality of g.

Suppose that g has more than two non-parallel translation actions Let v_a, v_b, and v_c be the velocity vectors in the world frame of three translation actions of g. By Equation 3.3, we know that the projection of v_a, v_b, and v_c onto the control line must be the Hamiltonian value H. Let v_{a_y}, v_{b_y}, and v_{c_y} be the projection of v_a, v_b, and v_c

onto the norm of the control line. By the Pigeonhole Principle, we know that at least two of v_{a_y}, v_{b_y}, and v_{c_y} have the same sign.

Without loss of generality, assume that v_{a_y} and v_{b_y} have the same sign; let their durations be t_a and t_b respectively; see Figure 3.3. If $v_{a_y} = v_{b_y}$, then the velocity vectors v_a and v_b are identical. This contradicts the assumption that v_a and v_b are non-parallel. If $v_{a_y} \neq v_{b_y}$, then without loss of generality, we assume $|v_{a_y}| > |v_{b_y}|$. Since the projections of v_a and v_b onto the control line are the same, we can remove the actions of v_b from g and increase the duration of v_a to $t_a + \frac{t_b|v_{b_y}|}{|v_{a_y}|}$. The resulting strategy still reaches the goal.

Now, we analyze the cost of the resulting trajectory. Let u be the control corresponding to the translation vector v_b. Let u_p and u_q be the controls before and after u in the trajectory. The resulting trajectory has a cost smaller than g's cost by $\frac{t_b|v_{b_y}|}{|v_{a_y}|} - t_b - C(u_p, u) - C(u, u_q) + C(u_p, u_q)$, which is strictly larger than zero. Hence, the resulting trajectory has smaller cost and contradicts the optimality of g. □

Singular, TGT, and regular trajectories

Same as the classification under the cost-free-switch model (Furtuna 2011), we classify control line trajectories into four classes: singular, TGT, generic, and regular.

A control line trajectory is called *singular* if there exists a non-zero measure interval along the trajectory that multiple controls have the same Hamiltonian value within the interval.

As an extension of a result under the cost-free-switch model (Furtuna 2011), any singular trajectory under the costly-switch model contains exactly one translation with velocity vector parallel to the control line, or contains exactly one switch from one translation to another translation. Hence, by Equation 3.3, the Hamiltonian value is either equal to the velocity of the only translation, or can be computed from the pair of consecutive translations. Since the control set U is a given finite set, the set of all possible Hamiltonian values for singular trajectories is finite.

A control line trajectory is called *generic* if the trajectory is not singular. For generic trajectories, switching between two translations can not occur, since switching between two translations only happens for singular trajectories. A generic trajectory is further called TGT if both the first control and the last control are translations, and *regular* otherwise. For a TGT trajectory, when the start configuration and goal configuration are given, we can obtain the Hamiltonian value analytically, using methods similar to the method used under the cost-free-switch model (Furtuna 2011). For regular trajectories, since we cannot determine the Hamiltonian value analytically, we use a numerical method to find an approximately optimal trajectory.

3.3.2 Whirl trajectories

For whirl trajectories, Equation 3.3 only implies that all angular velocities are equal. We also extend the result under the cost-free-switch model (Furtuna 2011) to the costly-switch model.

First, we show that in order to compute an optimal whirl trajectory, it suffices to consider a smaller subclass, called *two-stage whirl trajectories*:

(a) Move the last rotation center to the correct position in the goal configuration using the minimum cost.

(b) Rotate around the last rotation center until the goal configuration is achieved.

Then, we extend the results under the cost-free-switch model (Furtuna 2011) to obtain the following two theorems.

Theorem 3.3. *For any kinematic planar rigid body under the costly-switch model, among all whirl trajectories, there exists one two-stage trajectory with the minimum cost.*

Proof. Let T_1 and T_2 be the durations corresponding to the first and the second stage respectively. Let T_f be the duration of an optimal trajectory. Since T_f is the duration of an optimal trajectory, $T_f \leq T_1+T_2$. Moreover, since an optimal trajectory needs to place the last rotation center in the correct position, $T_f \geq T_1$. Since T_2 is strictly less than 2π, we have $T_f \leq T_1 + T_2 < T_f + 2\pi$. For any two admissible whirl trajectories, the difference of total durations must be a multiple of 2π. Therefore, T_f must equal $T_1 + T_2$. □

Theorem 3.4. *Any two-stage trajectory must satisfy the following property: there exist three constants $H_\omega > 0$, k_α, and k_β, such that for any control u_i with the instantaneous velocity (v_x, v_y, ω) in the world frame when u_i,$1 \leq i < n$, is applied at configuration (x, y, θ), we have*

$$k_\alpha v_x + k_\beta v_y = H_\omega, \text{ where } k_\alpha^2 + k_\beta^2 = 1. \tag{3.4}$$

Control direction interpretation for two-stage trajectories.

For a two-stage trajectory, we define its *control direction* as a line heading (k_α, k_β) through the rotation center of the first control. By Equation 3.4, all rotation centers except the last one should have the same signed distance to this line. Since the first

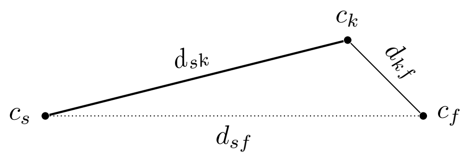


Figure 3.4: Illustration of proof of Theorem 3.5. The thick line is the control direction where all rotations except the last one are on this line.

rotation is on the control direction, all rotation centers except the last one are on a line that is parallel to the control direction.

We extend the result under the cost-free-switch model (Furtuna 2011) to obtain the following theorem.

Theorem 3.5. *For the costly switch model, consider a two-stage trajectory* $(\mathbf{u}, \mathbf{t})$ *with* $u_s = u_1, ..., u_{n-1} = u_k, u_n = u_f$. *Let* c_s, c_k, *and* c_n *be the rotation centers of* u_1, u_{n-1}, *and* u_n *respectively. Let* d_{sf} *be the distance between* c_s *and* c_f. *Let* d_{kf} *be the distance between the rotation centers of* u_k *and* u_f *in the robot frame. Let* l_i *be the distance between the rotation centers of* u_i *and* u_{i+1} *in the robot frame. Let* $d_{sk} = \sum_{i=1}^{n-2} l_i$. *We have*

$$d_{sk} \in [|d_{sf} - d_{kf}|, d_{sf} + d_{kf}]. \tag{3.5}$$

Proof. By the geometric interpretation of Theorem 3.4, since all rotation centers from u_1 to u_{n-1} are on the same line, we know that d_{sf}, d_{kf}, and d_{sk} form a triangle; see Figure 3.4. Moreover, any sequence of controls $\mathbf{u}$ that all controls have the same angular velocity and satisfy Equation 3.5 can form a two-stage trajectory. □

3.3.3 Taxonomy of optimal trajectories

We summarize the taxonomy of optimal trajectories as Figure 3.5.

Since the Hamiltonian values for whirl, TGT, and singular trajectories can be

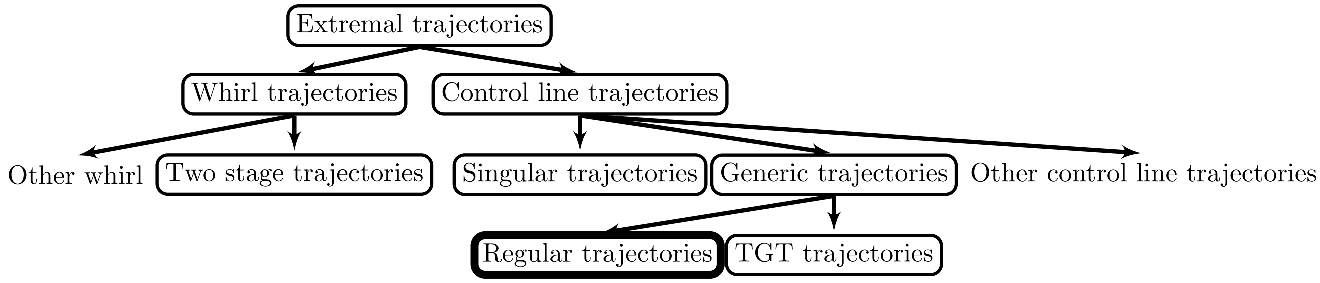


Figure 3.5: Taxonomy of optimal trajectories. Each node corresponds to a type of optimal trajectories; each leaf node without border is not necessary for optimality. All leaf nodes with single border can be solved exactly. For the leaf node with double border, regular trajectories, we provide a search algorithm that can find a trajectory with cost arbitrarily close to the cost of optimal trajectories.

determined, the problems of finding optimal trajectories in these three classes is equivalent to finding an optimal sequence of controls, a discrete search problem. For these three classes, we have designed three different A* search algorithms to find candidate optimal trajectories by searching over discrete trajectory structures.

The problem of finding optimal regular trajectories has two ingredients: one is finding the Hamiltonian value H, which is a continuous variable, and another one is finding the sequence of controls, chosen from a finite set.

3.4 The bench mover's problem

In this section, as a working example, we demonstrate how to use necessary conditions of optimal trajectories to solve the bench mover's problem exactly.

3.4.1 Model and trajectory types

Consider a park bench with length 2. Let q_s be the start configuration and $(0, 0, 0)$ be the goal configuration. Figure 3.6 gives an example with start configuration

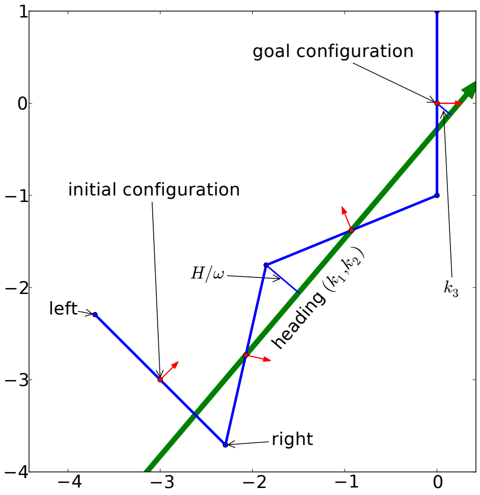


Figure 3.6: Optimal trajectory for start configuration $(-3, -3, \pi/4)$ with switching cost 1, where arrow represents the orientation of the bench. Thick line denotes the control line for this trajectory.

$(-3, -3, \pi/4)$. Let the reference point be the center of the bench, $(0, 0)$ in the robot frame. There are two rotation centers: the left rotation center, $(0, 1)$ in the robot frame, and the right rotation center, $(0, -1)$ in the robot frame.

Let $l^+ = (1, 0, 1)$ and $l^- = (-1, 0, -1)$ be the controls rotating around the left rotation center with different angular velocities. Similarly, let $r^+ = (-1, 0, 1)$ and $r^- = (1, 0, -1)$ be the controls rotating around the right rotation center with different angular velocities. The control set can be denoted as $U = L \cup R$, where $L = \{l^+, l^-\}$ and $R = \{r^+, r^-\}$. For two controls $u, u' \in U$, the cost of switching from u to u' is a constant c.

Since we can determine optimal durations for a control sequence with length smaller than three easily, we focus on the case where the length of the control sequence is at least three.

Based on the taxonomy of optimal trajectories, there are three broad types of trajectories:

(a) Whirl: trajectories for which $k_x = k_y = 0$. All controls in the trajectory must have the same angular velocity.

(b) Alternating sign: the control sequence contains controls alternating between l^+ and r^- or alternating between l^- and r^+.

(c) Mixed: the control sequence contains controls alternating between L and R but not strictly alternating signs.

Our basic approach, given a start configuration, is to compute an optimal trajectory of each of the three types, and then to compare to find the minimum. The following sections will demonstrate how to find an optimal trajectory for each type. For computing optimal trajectories of types b and c, an upper bound on the number of control actions in the trajectory is required; this bound may be found by considering the cost of an optimal whirl trajectory.

3.4.2 Whirl trajectories

For whirl trajectories, all rotation centers except possibly the last one are on the same line; see Figure 3.7. This section will show how this fact can be used to identify a minimum-cost whirl trajectory.

When the first control and the last control are fixed, the distance between their rotation centers is determined. Since the length of the bench is two and controls alternate between L and R, for any two consecutive controls, the distance between their rotation centers is two. Thus, in order to reach the goal, there is only one choice

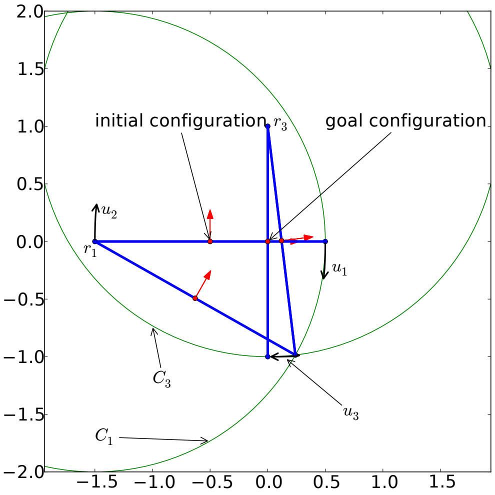


Figure 3.7: Whirl trajectory with start configuration $(-0.5, 0, \pi/2)$. All rotation centers except the last one are on the same line. This is the optimal trajectory for this start configuration with switching cost 1.

of the length of the control sequence; see Figure 3.8. Hence, if we fix the first control and the last control, only two whirl trajectories exist. Since there are only four choices for the first control and each has two choices for the last control, we can enumerate all possible pairs of first and last controls for whirl trajectories; see Figure 3.8.

Now, we show how to determine the durations of the all controls in a whirl trajectory. Fix the first control u_1 and the last control u_n, with rotation centers r_1 and r_n respectively. Since the last control is fixed, the second to the last control u_{n-1} is also fixed and its rotation center r_{n-1} should be on a circle C_n centered at r_n with a radius of two.

Since rotation centers r_i, $1 \leq i < n$, are on the same line, the distance P from r_1 to r_{n-1} is determined in the following way. Let D be the distance between the first and the last rotation centers. If $u_1 = u_n$ $(u_1 \neq u_{n-1})$, then P is multiple of

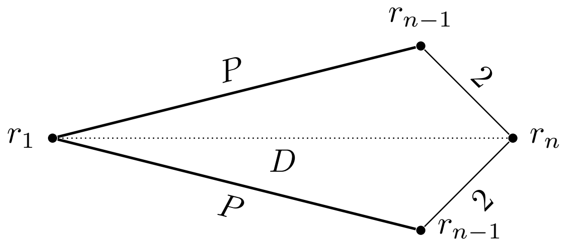


Figure 3.8: When the first control and the last control are fixed, the rotation centers r_1 and r_n are fixed as well. The distance between r_1 and r_n is D. Since all rotation centers except for the last one is on the same line and distance between two consecutive rotation centers is two, the distance between r_1 and r_{n-1}, L, is a multiple of two. Since the distance between r_{n-1} and r_n is two, there is only one choice of P and two symmetric choices of locations of r_{n-1}

four plus two. Otherwise, P is multiple of four. Since the diameter of C is four and the difference between any choices is multiple of four, $P = 4\lceil (D-4)/4 \rceil + 2$ when $u_1 = u_n$, otherwise $P = 4\lceil (D-2)/4 \rceil$.

After we determine P, we can find a circle C_1 centered at r_1 with radius P. The circle C_1 intersects with C_n at most two points and these points are possible locations of r_{n-1}. When the location of r_{n-1} is fixed, the durations for all controls can be determined easily.

3.4.3 Alternating sign trajectories

Since all angular velocities have the same absolute value, all rotation centers must have equal distance to the control line; see Figure 3.6. We will show that in order to find the best alternating sign trajectory, it suffices to enumerate a finite set of alternating sign trajectories.

The idea is as follows:

(a) There are only four choices of first control u_1 in U.

(b) When the first control is fixed, there are only two choices for the last control: $u_n = u_1$ if n is odd, or u_n has a different rotation center with a different angular velocity from u_1.

(c) When the first control and the last control are fixed, we show that there are at most two possible values of the Hamiltonian.

(d) When the first control and the last control are fixed, and a Hamiltonian value is given, we show that there are at most two control lines.

(e) When the first control and the last control are fixed, a control line and the length of the control sequence are given, we show that that there are at most two choices of durations of the alternating sign trajectory.

Since the length of the control sequence is bounded by the cost of a feasible trajectory, we can determine the best alternating trajectory by enumerating all possible first control, last control, Hamiltonian values, control lines, length of control sequence, and durations of controls in finite time. We show (c), (d), and (e) in the following sections.

Determining the Hamiltonian value H

Let r_1 and r_n be the first rotation center and the last rotation center, separated by distance D. If n is odd, we have $D = (2n-2)\sqrt{1-H^2}$ and $H = \sqrt{1 - \frac{D^2}{4(n-1)^2}}$. In this case, when $0 < D^2 \leq 4(n-1)^2$, control lines exist and we obtain a positive value of $H \leq 1$. When n is even, let X be $(2n-4)\sqrt{1-H^2}$, D^2 will be $X^2 + \sqrt{1-H^2} + 4$. Consequently, we have $D^2 = 4n(n-2)(1-H^2) + 4$ and $H = \sqrt{1 - \frac{D^2-4}{4n(n-2)}}$. In this case, when $4 < D^2 \leq 4n(n-2)+4$, control lines exist and we obtain a positive value of $H \leq 1$.

Determining control lines

After we determine the Hamiltonian value H, we determine control lines, and each control is represented by a tuple (k_x, k_y, k_θ). Since $k_x^2+k_y^2 = 1$, we can use $(\cos\varphi, \sin\varphi)$ to represent (k_x, k_y). For one Hamiltonian value, H, there are two possible control lines. We determine (φ, k_θ) in a similar way as the way used under the cost-free-switch model (Furtuna 2011). Let $r'_{1_x} = r_{1_x} u_{1_\omega}, r'_{1_y} = r_{1_y} u_{1_\omega}, r'_{n_x} = r_{n_x} u_{n_\omega}$, and $r'_{n_y} = r_{n_y} u_{n_\omega}$. Let $d_x = r'_{1_x} - r'_{n_x}$ and $d_y = r'_{1_y} - r'_{n_y}$. Let (α, β) be $(\text{atan}(d_x, d_y), \pi/2)$ if the first control and the last control have the same angular velocity; otherwise $(\text{atan}(d'_x, d'_y), \text{acos}(\frac{H}{\sqrt{{d'_x}^2+{d'_y}^2}}))$, where $d'_x = r'_{n_x} + d_x/2$ and $d'_y = r'_{n_y} + d_y/2$. Then, $\varphi = -\alpha \pm \beta$ and $k_\theta = \frac{H + r'_{n_x}\sin\varphi - r'_{n_y}\cos\varphi}{u_{n_\omega}}$.

Determining durations

For a given control line $L = (k_x, k_y, k_\theta)$ with a Hamiltonian value H, we can determine the durations of all controls as follows. By Theorem 3.1, all rotation centers have the same distance H to L. This strongly constraints the duration of the first control, since we must apply the first control until the rotation center of the second control has distance H to L. When the first control is applied, the motion of the rotation center of the second control is a circle, so there are only two locations for the second rotation center. And we can determine the duration for all other controls in the same manner and the details are described below.

We determine the location of r_2 for u_2 as follows. By Theorem 3.1, all rotation centers have the same distance H to L. For two consecutive rotation centers, their distance must be 2, the length of the bench. Hence, the angle β between the vector $r_2 - r_1$ and L can only have two possible values: $\mathrm{asin}(H)$ and $\pi - \mathrm{asin}(H)$ if $u_{1_\omega} > 0$, otherwise $\pi + \mathrm{asin}(H)$ and $2\pi - \mathrm{asin}(H)$. For a fixed β, we can determine t_1.

For u_2, since we switch to u_2 at angle β with respect to L and $u_3 = u_1$, we also can switch to u_3 immediately with $t_2 = 0$. Since this null control will be examined by another control sequence, we ignore this choice. Consequently, we have only one choice of t_2. Similarly, all controls u_2 to u_{n-1} have the same duration. This duration will be either $2\mathrm{acos}(H)$ (if $\beta \in [\pi/2, \pi/2]$) or $2\pi - 2\mathrm{acos}H$ (otherwise). The duration t_n can be determined based on t_1 to t_{n-1} based on the constraint of reaching the goal configuration.

3.4.4 Mixed trajectories

In this section, we will show that in order to find the best mixed trajectory, it suffices to enumerate a finite set of mixed trajectories. The argument is similar to the previous section: when the first control and the last control are determined, there are at most finite choices of Hamiltonian values, control lines, length of the control sequences, and durations of all controls. The difference is that when the first control and the last control are fixed, there may be more than two possible values of the Hamiltonian, but the number of choices is still finite.

First, we show that for any mixed trajectory, we can rearrange the controls without changing the cost, such that the controls can be divided into two parts: the prefix of the control sequence has controls with the same angular velocity and the suffix has controls with alternating angular velocity. Consider a mixed trajectory that contains

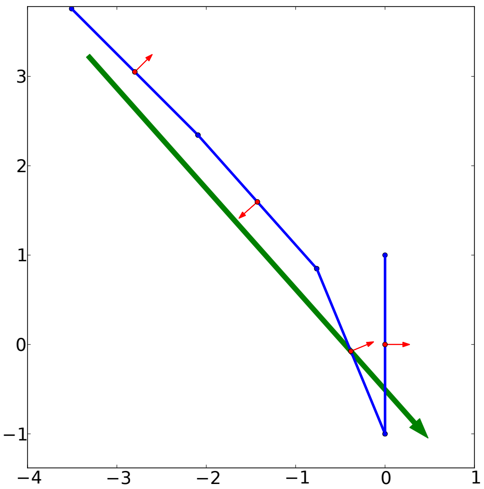


Figure 3.9: Mixed trajectory with start configuration $(-2.8, 3.05, \pi/4)$. First two controls have the same angular velocity and hence they are collinear and parallel to the control line. This is the optimal trajectory for this start configuration with switching cost 1.

two consecutive controls, u and u', with the same angular velocity. By Theorem 3.1, these two controls' rotation centers, r and r', have equal distance to the control line, L, and are on the same side of the control line. Hence, the line through r and r' is parallel to L; see Figure 3.9. Consequently, if there are three consecutive controls with the same angular velocity, then the duration of the second one must be π.

For a mixed trajectory, subsequences of controls with the same angular velocity may appear anywhere in the control sequence. However, it is always possible to rearrange the controls in the control sequence without changing the cost, such that the prefix of the control sequence has controls with the same angular velocity and the suffix has controls with alternating angular velocity. Hence, we only consider control sequences that can be decomposed into two parts in this way.

Now, we show that there are only finite choices of Hamiltonian values. Let n

be the length of the control sequence, $n_w < n$ be the number of controls with the same sign, and m be $n - n_w$. Let D be the distance between the first rotation center and the last rotation center. When m is even, H satisfies $|D - 2(n_w - 1)| = 2m\sqrt{1 - H^2}$ or $D + 2(n_w - 1) = 2m\sqrt{1 - H^2}$. Hence, $H = \sqrt{1 - \frac{(D-2(n_w-1))^2}{4m^2}}$ or $H = \sqrt{1 - \frac{(D+2(n_w-1))^2}{4m^2}}$. We can obtain at most two possible positive $H \leq 1$ values.

When m is odd, H satisfies $D^2/4 = (m^2-1)(1-H^2)+2m(n_w-1)\sqrt{1 - H^2}+n_w^2 - 2n_w + 2$ or $D^2/4 = (m^2 - 1)(1 - H^2) - 2m(n_w - 1)\sqrt{1 - H^2} + n_w^2 - 2n_w + 2$. Hence, $1 - H^2 = |\frac{-b\pm\sqrt{b^2-4ac}}{2a}|$, where $a = m^2 - 1$, $b = 2m(n_w - 1)$, and $c = n_w^2 - 2n_w + 2$. We can obtain at most two possible positive Hamiltonian values $H \leq 1$ values. Since the ranges of n, n_w, and m are finite sets, the number of possible Hamiltonian values is also finite.

3.5 Outline of an algorithm for finding optimal trajectories for kinematic planar rigid-body systems

In the previous section, we showed how to find optimal trajectories for a specific system, the bench mover's problem, exactly. However, for more complex systems, deriving an analytical result is challenging, since we not only need to determine the optimal Hamiltonian value but also the optimal control sequence. In this section, we design an algorithm to find (approximately) optimal trajectories for all rigid-body systems.

First, we want to find an upper bound of number of switches for optimal trajectories. The upper bound limits the length of control sequence that the algorithm needs to enumerate. We use the general planner for kinematic planar rigid-body robot (Fur-

3.5 Outline of an algorithm for finding optimal trajectories for kinematic planar rigid-body systems

Algorithm 1: Outline of the algorithm

```
Algorithm CostlySwitchOptimalMotionPlanner()
    $C \leftarrow$ feasible trajectories found by any planner.
    $B \leftarrow$ upper bound of number switches.
    // Find best two-stage trajectories; see Section 3.6
    foreach 3-tuple of controls $(u_s, u_k, u_f)$ with the same angular velocity. do
        $C = C\cup$ best two-stage trajectory $(\mathbf{u}, \mathbf{t})$ with $u_1 = u_s$, $u_{n-1} = u_k$,
         $u_n = u_f$.
    end
    // Find best TGT trajectories; see Section 3.8
    foreach 2-tuple of translation controls $(u_s, u_f)$ do
        $C = C\cup$ best TGT trajectory $(\mathbf{u}, \mathbf{t})$ with $u_1 = u_s$ and $u_n = u_f$.
    end
    // Find best singular trajectories; see Section 3.9
    foreach 2-tuple of controls $(u_s, u_f)$ do
        foreach singular value $H$ do
            $C = C\cup$ best singular trajectory $(\mathbf{u}, \mathbf{t})$ with $u_1 = u_s$, $u_n = u_f$ and
             Hamiltonian value $H$.
        end
    end
    // Find best regular trajectories; see Section 3.10
    foreach 2-tuple of controls $(u_s, u_f)$ do
        foreach interval $I$ of Hamiltonian values not including singular values
         do
            $C = C\cup$ approximately best regular trajectory $(\mathbf{u}, \mathbf{t})$ with $u_1 = u_s$
             and $u_n = u_f$ in the interval $I$.
        end
    end
    return the optimal trajectory in $C$
```

tuna 2011) to obtain a feasible trajectory from q_s to q_f with cost M. Then, we can get an upper bound of number switches as $B = \lceil M/c_{\min} \rceil$, where $c_{\min}$ is the minimum switching cost.

By the taxonomy of optimal trajectories, we know that it is sufficient to find best two-stage, TGT, singular, and regular trajectories. For each of these four classes, we design algorithms separately in Section 3.6 to Section 3.10.

Since two-stage trajectories are whirl trajectories and all other three classes are control line trajectories, we explain how to find best two-stage trajectories first in Section 3.6. Then, we introduce several properties for control line trajectories in Section 3.11. Finally, we show how to find best TGT trajectories in Section 3.8, singular trajectories in Section 3.9, and regular trajectories in Section 3.10.

The idea of finding best trajectories within each class is quite similar: decompose the problem into several sub-problems by enumerating the first control, the last control, and possibly the second to the last control (for two-stage trajectories only). For two-stage, TGT, and singular trajectories, the sub-problems are discrete optimization problems so that we use an A* search algorithm to find best trajectories exactly. For regular trajectories, the sub-problem is a mixed non-linear optimization problem; we combine Lipschitzian optimization techniques with an A* search algorithm over discrete trajectory structures to determine approximately best trajectories. We give the outline of the algorithm in Algorithm 1.

3.6 Finding best two-stage whirl trajectories

Recall that a two-stage trajectory is a trajectory such that all controls have the same angular velocity and all rotation centers except the last one are on the same line. Our approach is to enumerate all 3-tuple of controls (u_s, u_k, u_f) with the same angular velocity. For each $(u_s, u_k, u_f) \in U^3$, we determine a best two-stage trajectory $(\mathbf{u}, \mathbf{t})$, subject to $u_1 = u_s$, $u_{n-1} = u_k$, and $u_n = u_f$, where the number of controls, n, can be determined during the search. Then, we pick the best trajectory among all best two-stage trajectories with respect to all 3-tuples of controls.

Fix a 3-tuple of controls (u_s, u_k, u_f) with the same angular velocities. Let U' be the reduced control set that all controls in U' have the same angular velocity as u_s. We want to find a best two-stage trajectory $(\mathbf{u}, \mathbf{t})$, subject to $u_s = u_1, ..., u_{n-1} = u_k, u_n = u_f$ and $u_i \in U'$ for all i.

Our method is to incrementally build sequences of controls that could possibly satisfy Equation 3.5 by using A* search. Each state is a sequence of controls $g = (u_1 = u_s, \ldots, u_{h-1} = u_k, u_h = u_f)$, where all controls have the same angular velocity. For a state $g = (u_1, \ldots, u_h)$, the duration of each control u_i, $1 < i < h-1$ is fully determined, since all rotation centers except the last one are on the same line. Hence, we use the summation of the switching costs and the durations for each control u_i, $1 < i < h-1$ as path cost. The neighbors of a state $g = (u_1, \ldots, u_h)$ form a set $\{(u_1 = u_s, \ldots, u_{h-1}, u', u_h = u_k, u_{h+1} = u_f) \mid u' \in U'\}$.

A state $g = (u_1, \ldots, u_h)$ reaches the goal if $(u_1, \ldots, u_h, u_k, u_f)$ satisfies Equation 3.5. When a state reaches the goal, we can solve for the duration of u_1 and u_h exactly with at most two solutions.

In order to speed up the A* search, we need an admissible heuristic. Let d_{sf} be the distance between the rotation center of u_s at q_s and u_f at q_f. For a state $g = (u_1, \ldots, u_h)$, let $l_g = \sum_{i=1}^{h-2} D(u_i, u_{i+1})$, where $D(u, u')$ is the distance between the rotation centers of u and u' in the robot frame. That is, l_g is the possible value of d_{sk} for the state in Equation 3.5. In order to satisfy Equation 3.5, we know that I_g must be at least $|d_{sf} - D(u_k, u_f)|$. Hence, we use $|d_{sf} - D(u_k, u_f)| - l_g$ as an admissible heuristic.

3.7 Properties of control line trajectories

In this section, we briefly describe several properties of control line trajectories; additional details are in Section 3.11 and Section 3.12.

Recall that a control line trajectory is an extremal trajectory with constants H, k_x, k_y, and k_θ such that $k_x^2 + k_y^2 = 1$. When the start configuration q_s, the first control u_s, the final configuration q_f, the last control u_f, and the Hamiltonian value H are given, the control line can be constructed explicitly:

Theorem 3.6. *For a given start configuration q_s, given goal configuration q_f, first control u_s, and the last control u_f, we can determine an interval of Hamiltonian values $I = (0, H_u)$, such that there exists mappings, $L_1(H)$ and $L_2(H)$, from Hamiltonian values in I to control lines.*

A *singular interval* is an interval of time within which at every time more than one control gives the same value for the Hamiltonian. Except within singular intervals, knowing the configuration of the rigid body, the location of the control line, and the current and next controls, essentially tells when the next switch will be.

Theorem 3.7. *Given a control line L and a non-singular interval, the duration of applying a control u from a configuration q until switching to another control u' has at most two possibilities such that the resulting motion can be a subtrajectory of a control* *line trajectory corresponding to the control line L. Moreover, these two possibilities can be determined analytically.*

Singular trajectories can only occur at particular critical values of the Hamiltonian. Except at these critical values, knowing the Hamiltonian value H and three consecutive controls is sufficient to compute the duration of the middle control. The following theorem will be proved in Section 3.12:

Theorem 3.8. *Let u, u', and u'' be three consecutive controls in a control line trajectory. Given k_x, k_y, and a non-critical Hamiltonian value H, the duration of u' has at most two possibilities and can be determined analytically without knowing the configuration.*

Finally, for a given sequence of controls and a control line L, there is a way to determine the duration of each control when we are given the start configuration q_s and the final configuration q_f:

Theorem 3.9. *Let $\mathbf{u} \in U^n$ be a sequence of controls. Let $\mathbf{b} \in \{1, 2\}^{n-1}$ be a sequence of numbers. For a given control line L with a non-critical Hamiltonian value H, start configuration q_s, and final configuration q_f, the duration of each control is fully determined by $\mathbf{u}$ and $\mathbf{b}$.*

By Theorem 3.6, the control line can be parametrized by the Hamiltonian value H. Suppose that the mapping from the Hamiltonian values to control lines is fixed. Given a sequence of control $\mathbf{u}$ and selectors, since the duration is fully determined by the control line L, the durations can also be parametrized by the Hamiltonian value H.

In order to find best trajectories corresponding to a control line L of known location, with given initial and final configurations of the rigid body, it suffices to search over only a finite set of trajectories. Since for a control line L, the number of corresponding trajectories is finite, if there is only a finite number of control lines as well, then we can possibly find best trajectories exactly by enumerating all possible control lines and corresponding trajectories. As it turns out, the number of control lines for TGT and singular trajectories are finite and we also can enumerate them. Hence, for TGT and singular trajectories, we can find best trajectories exactly. However, for regular trajectories, we do not have a method to reduce the number of control lines to a finite set and that's why we only find approximately best regular trajectories.

3.8 Finding best TGT trajectories

Remember that a TGT trajectory is a control line trajectory for which both the first control and the last control are non-parallel translations. Our approach is to enumerate all pairs of controls $(u_s, u_f) \in U^2$, where both u_s and u_f are translations. For each (u_s, u_f), we determine a best TGT trajectory $(\mathbf{u}, \mathbf{t})$, subject to $u_1 = u_s$, and $u_n = u_f$. Finally, we pick the best TGT trajectory among all best TGT trajectories with respect to all pairs of controls.

Fix a pair of translation controls (u_s, u_f). Since the start configuration q_s and the final configuration q_f are given, we can compute the velocity vectors v_s and v_f for applying u_s at q_s and u_f at q_f respectively. Then, we substitute v_s and v_f in Equation 3.3 and we get a system of linear equations. Since we know $k_x^2 + k_y^2 = 1$ and $H > 0$, we can solve k_x, k_y, and the Hamiltonian value H exactly.

Our method is to incrementally build sequences of controls that could possibly satisfy Equation 3.3 by using A* search. Each state $g = (\mathbf{u}, \mathbf{t})$ is a pair of sequences of controls and durations, where $|\mathbf{u}| = |\mathbf{t}| = h$, $u_1 = u_s$, and $u_h = u_f$. Each state also satisfies that each t_i, $1 < i < h$, is computed according to Theorem 3.7, but t_1 and t_h are undefined. We use the summation of the switching costs and the durations t_i, $1 < i < h$ as path cost. The neighbors of a state $g = (\mathbf{u}, \mathbf{t})$ form a set $\{((u_1, \ldots, u_{h-1}, u', u_{h+1} = u_f), (t_1, \ldots, t_{h-1}, t', t_{h+1})) \mid u' \in U$ and t' is computed according to Theorem 3.8$\}$.

We can test whether a state $g = (\mathbf{u}, \mathbf{t})$ reaches the goal as follows: For a state g, since all t_i, $1 < i < h$, are determined, the displacement, $(\delta_x, \delta_y, \delta_\theta)$, in the configuration space of the sub-trajectory $u_2, \ldots, u_{h-1}$ can be computed. If $q_{s,\theta} + \delta_\theta \neq q_{f,\theta}$, then since u_s and u_f are translations, g cannot reach the goal. Otherwise, since we know the velocity v_s and v_f at the start configuration and the final configuration respectively, we solve the following system of linear equations to get durations $t_1 = t_s$ and $t_h = t_f$ for u_s and u_f.

$$q_{s,x} + v_{s,x} t_s + \delta_x + v_{f,x} t_f = q_{f,x} \tag{3.6}$$

$$q_{s,y} + v_{s,y} t_s + \delta_y + v_{f,y} t_f = q_{f,y} \tag{3.7}$$

In order to speed up A* search, we need to design an admissible heuristic. A state $g = (\mathbf{u}, \mathbf{t})$ can reach the goal if and only if the change of orientation, δ_θ, equals $q_{f,\theta} - q_{s,\theta}$. Hence, we can use the difference $|q_{f,\theta} - q_{s,\theta} - \delta_\theta|$ as an admissible heuristic.

3.9 Finding best singular trajectories

Remember that a control line trajectory is called *singular* if there exists a non-zero measure interval along the trajectory that multiple controls have the same Hamiltonian value within this interval. Furthermore, the number of singular Hamiltonian value is a finite set when U is given. Our approach is to enumerate all pairs of controls $(u_s, u_f) \in U^2$ and singular Hamiltonian values, H, where one of u_s and u_f is a rotation. For each (u_s, u_f) and H, we determine a best singular trajectory $(\mathbf{u}, \mathbf{t})$, subject to $u_1 = u_s$, $u_n = u_f$ and the Hamiltonian value is H. Finally, we pick the best singular trajectory among all best singular trajectories with respect to all pairs of controls and singular Hamiltonian values.

Fix a pair of controls (u_s, u_f) and a singular Hamiltonian value H, where one of u_s and u_f is a rotation. Based on u_s, u_f, and H, we can construct two control lines according to Theorem 3.6. Fix a control line L. Our method is to incrementally build sequences of controls that could possibly satisfy Equation 3.3 by using a bidirectional A* search.

In the bidirectional A* search, there are two different states, S and F, denoting the state grow from q_s and q_f(in reverse) respectively. Each state, $g = (\mathbf{u}, \mathbf{t})$, is a pair of sequences of controls and durations, where $|\mathbf{u}| = |\mathbf{t}| = h$. If $g \in S$, then $u_1 = u_s$ and each t_i, $1 \leq i < h$, is computed according to Theorem 3.7 assuming the trajectory starts at q_s. If $g \in F$, then $u_h = u_f$ and each t_i, $1 < i \leq h$, is computed according to Theorem 3.7 assuming the trajectory is built from q_f backwards. In this way, each state has exactly one undefined duration and we use the summation of the switching cost and the defined durations as path cost.

The neighbors of a state $g = (\mathbf{u}, \mathbf{t}) \in S$ form a set $\{((u_1, \ldots, u_h, u'), (t_1, \ldots, t_h, t')) \mid$

where $u' \in U$, and t_h is computed according to Theorem 3.7$\}$. We define the neighbors of a state $g \in F$ symmetrically.

For a pair of states $g = (\mathbf{u}, \mathbf{t}) \in S$ and $g' = (\mathbf{u}', \mathbf{t}') \in F$, where $|\mathbf{u}| = h$ and $|\mathbf{u}'| = h'$, if both u_h and u'_1 are translations, then we can combine the two states to construct a feasible singular trajectory by solving a system of linear equations similar to Equation 3.6 and Equation 3.7. In order to speed up the search by using bidirectional A*, we design an admissible heuristic as follows. For a state $g = (\mathbf{u}, \mathbf{t}) \in S$, let q_g be the configuration that start from q_s and apply all u_i with duration t_i in order for all $1 \leq i < h$. The distance between q_g and q_f is the lower bound of the length of the trajectory from the state to the goal. Hence, we use the Euclidean distance between q_g and q_f divide by the maximum velocity as an admissible heuristic. For a state $g \in F$, we design a similar heuristic.

3.10 Finding best regular trajectories

Remember that a regular trajectory is a generic trajectory either starting or ending with a rotation. Our approach is to enumerate all pairs of controls $(u_s, u_f) \in U^2$, where one u_s of u_f is a rotation. Unlike TGT and singular trajectories, the number of potential control lines for regular trajectories is uncountably infinite. Hence, we use Lipschitzian optimization techniques to determine the best Hamiltonian value and its corresponding control line.

In order to use Lipschitzian optimization techniques, we need to define a Lipschitzian optimization problem. However, even when we fix a pair of controls (u_s, u_f), the trajectories may behave differently with respect to the Hamiltonian value H. For some Hamiltonian values H, control u can switch to another control u', but the switch cannot happen for some other Hamiltonian values H', and this may destroy Lipschitz continuity. Therefore, for each (u_s, u_f), we partition the Hamiltonian values into several disjoint open intervals so that within each interval the change of trajectories with respect to the Hamiltonian value is Lipschitz continuous.

For each interval I, we use Lipschtzian optimization techniques to determine a best regular trajectory $(\mathbf{u}, \mathbf{t})$, subject to $u_1 = u_s$, $u_n = u_f$, and the corresponding Hamiltonian value $H \in I$. During this step, we need a method to determine a regular trajectory corresponding to a fixed control line L that approximately minimizes error and time. Similar to the idea we used for finding best TGT and singular trajectories, we use A* search to find a regular trajectory that approximately minimizes error and time. Finally, we pick the best regular trajectory among all best regular trajectories with respect to all pairs of controls and all intervals of the Hamiltonian values.

In the following sections, we will explain how to partition the Hamiltonian values, reduce the problem to a Lipschtzian optimization problem, and determine best regular trajectories corresponding to a fixed control line.

3.10.1 Partitioning the Hamiltonian values

Fix a pair of controls (u_s, u_f), where one of u_s and u_f is a rotation. We show how to partition the Hamiltonian values into several disjoint open intervals so that within each interval I, if u, u', and u'' are three consecutive controls in a control line trajectory for a Hamiltonian value $H \in I$, then u, u', and u'' will also be well-defined for another Hamiltonian value $H' \in I$.

According to Theorem 3.6, we have two continuous functions mapping from the Hamiltonian value to a control line. Consider one fixed mapping $L(H)$ of these two mappings. According to Theorem 3.8, for any three controls u, u', and u', if these three controls are consecutive in a regular trajectory, then the duration of u' can be determined by the Hamiltonian value H. Based on the calculation, we can figure out the range of the Hamiltonian values that the duration of u' is well defined and must be an interval. That is, for each triple $e = (u, u', u'') \in U^3$, we can determine the range of the Hamiltonian values I_e that the duration of u' is well-defined.

We collect the intervals I_e for all triple $e = (u, u', u'') \in U^3$ that the duration of u' is well-defined for all Hamiltonian values in I_e. Let S be the set of numbers containing all endpoints for all intervals I_e.

The set S partitions the domain of $L(H)$, determined by the parametrization and the control set, into several disjoint open intervals. We have the following theorem.

Theorem 3.10. *There exists a finite set of critical values of $\mathcal{R}$ that partition the Hamiltonian values into a finite set of open intervals, such that for each interval I, if u, u', and u'' are three consecutive controls in a control line trajectory for a Hamiltonian value $H \in I$, then u, u', and u'' will also be well defined for another Hamiltonian value $H' \in I$.*

3.10.2 Reduction to a Lipschitzian optimization problem

Fix a pair of controls (u_s, u_f), an interval I of Hamiltonian values constructed in Theorem 3.10, and a mapping $L(H)$ from Hamiltonian values to control lines, we pose a Lipschitzian optimization problem to solve for the Hamiltonian value H with time and position error at most ϵ, for any desired $\epsilon > 0$. Here, we briefly introduce Lipschitzian optimization.

The goal of global optimization is to find optimal solutions of constrained optimization problem even for non-linear, non-continuous problems. A function $f : \mathcal{R} \to \mathcal{R}$ is called *Lipschitz continuous* if there exists a constant $L \geq 0$, such that for all pairs x, y in the domain we have $|f(x) - f(y)| \leq L|x - y|$, where L is called the *Lipschitz constant*. Given a Lipschitz continuous function $f(x)$, the problem of finding a global minimum, $\min_x f(x)$, is called a *Lipschitzian optimization* problem. For Lipschitzian optimization problems, there exist efficient algorithms to find globally (approximately) optimal solutions with arbitrarily small error in finite time(Pintér 1996).

One efficient algorithm for solving Lipschitzian optimization problem is Piyavskii's algorithm (Piyavskii 1967). The idea of Piyavskii's algorithm is to iteratively subdivide a domain I into several intervals. For each interval, Piyavskii's algorithm determines the lower bound of the objective function based on Lipschitz constant, and decides whether to further subdivide this interval or disregard this interval based on the lower bound information. For any error bound $\epsilon > 0$, Piyavskii's algorithm is guaranteed to find a solution with additive an error at most ϵ within a finite number of iterations.

The Lipschitizian optimization for finding best regular trajectories is formulated as follows:

$$\begin{aligned} \min\ & c(L, \mathbf{u}, \mathbf{t}) \\ & d(L, \mathbf{u}, \mathbf{t}) = 0 \\ & L = L(H) \text{ for some } H \in I, \text{ and } (\mathbf{u}, \mathbf{t}) \text{ is a regular trajectory.} \end{aligned} \tag{3.8}$$

The function c is the cost function that we want to minimize, which is the cost of the trajectory $(\mathbf{u}, \mathbf{t})$. The function d is the constraint that we want to satisfy, which should be the minimum distance from the trajectory $(\mathbf{u}, \mathbf{t})$ to the goal. We let $c(L(H), \mathbf{u}, \mathbf{t}) = d(L(H), \mathbf{u}, \mathbf{t}) = \infty$ if the trajectory $(\mathbf{u}, \mathbf{t})$ does not correspond to the control line $L(H)$.

In order to apply Lipschitzian optimization techniques, we need to show that functions c and d are Lipschitz continuous with respect to the change of the Hamiltonian value H. That is, we want to show that when the Hamiltonian values changes, the resulting distance and cost functions are Lipschitz continuous with respect to the Hamiltonian values. This differs from TGT and singular trajectories that we ignore all trajectories not reaching the goal exactly. We will prove the following theorem in Section 3.12.

Theorem 3.11. *Let $I = (a, b)$ be an open interval of the partition of the Hamiltonian values. Let $\mathbf{u}$ be a fixed sequence of n controls. Let $t_i(H)$ be the duration for the u_i and $d_i(H)$ be the length of projection of the sub-trajectory corresponding to u_i onto the control line. For any $\delta > 0$, both functions $t_i(H)$ and $d_i(H)$ are Lipschitz continuous with respect to the Hamiltonian values $H \in (a, b - \delta)$ for all $1 \leq i \leq n$.*

Moreover, we need a method to find best regular trajectory corresponding to a control line L that approximately minimizes error and time, which will be explained in the next section.

3.10.3 Finding optimal trajectories for a fixed control line L

Fix a pair of controls (u_s, u_f) and a control line L, we use A* search to find an best regular trajectory corresponding to the control line L approximately minimizing error. If it is possible to reach the goal with error at most ϵ, the result will be a regular trajectory approximately reaching the goal with approximately minimum cost. If it is impossible to reach the goal with error at most ϵ, the result will be a regular trajectory approximately minimizing the distance to the goal.

Our method is to incrementally build sequences of controls and durations that could possibly satisfy Equation 3.3 by using A* search. Each state $g = (\mathbf{u}, \mathbf{t})$ is a pair of sequences of controls and durations, where $|\mathbf{u}| = |\mathbf{t}| = h$ and $u_1 = u_s$ and $u_h = u_f$. Each state also satisfies that each t_i, $1 \leq i \leq h$, is computed according to Theorem 3.8 with an additional constraint that $\sum_i t_i$ is capped by M, the cost of a feasible trajectory. The neighbors of a state $g = (\mathbf{u}, \mathbf{t})$ form a set $\{((u_1, \ldots, u_{h-1}, u', u_{h+1} = u_f), (t_1, \ldots, t_{h-1}, t', t_{h+1})) \mid u' \in U$ and t' is computed according to Theorem 3.8$\}$.

We use the summation of the switching cost and the duration t_i, $1 \leq i < h$ as path cost. Note that we did not use t_h as part of the path cost; since the duration of $u_h = u_f$ depends on u_{h-1}, the sum of the durations of a state g may be larger than the sum of the durations of a g's neighbor.

For a state $g = (\mathbf{u}, \mathbf{t})$, we define the distance as follows: Let q_g be the final configuration of applying controls $\mathbf{u}$ with durations $\mathbf{t}$. If u_f is a translation, the distance from the state g to the goal is the distance between q_g and q_f. If u_f is a rotation, then let r_f be the rotation center of applying u_f at q_f and r_g be the rotation center of applying u_f at q_g, and the distance from the state g to the goal is the distance between r_f and r_g.

When the distance from the state g to the goal is zero, then the state g is at the goal. The distance divided by the maximum velocity can therefore also serve as an admissible heuristic for A* search.

There is one difficulty here: for a state $g = (\mathbf{u}, \mathbf{t})$, the switch from u_{h-1} to u_h may be impossible due to the constraint on the Hamiltonian values. In this case, we just

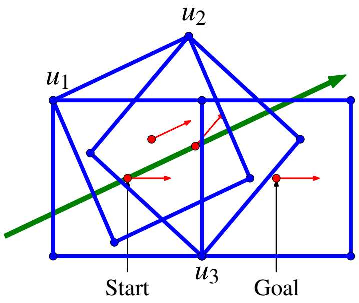


Figure 3.10: An approximately optimal trajectory derived using the described approximation algorithm for a refrigerator robot starting at (-2, 0, 0), with unit cost for switching between any pair of controls. The green line is the control line, and the u_i labels show the sequence of rotation centers.

pick the largest index i such that u_i can switch to u_{i+1} and let q_g be the configuration at which u_i switches to u_{i+1}. Then, use the Euclidean distance between q_g to q_f as the distance from the state g to the goal. In this case, we only use the distance to guide the search but will not use the trajectory in state g as a result, since the trajectory is infeasible.

3.11 Further properties of control line trajectories

In this section, we show several properties of control line trajectories. Remember that a trajectory $(\mathbf{u}, \mathbf{t})$ is called *extremal*, if there exist four constants $H > 0$, k_x, k_y, and k_θ, such that Equation 3.3 is satisfied. When the start configuration q_s, the first control u_s, the final configuration q_f, the last control u_f, and the Hamiltonian value H are given, the control line can be constructed explicitly. Furthermore, when a control line L is fixed, we can show that the trajectory along the control line has only finite number of possibilities.

3.11.1 Notation

For a given configuration $q = (x, y, \theta)$, there is a corresponding transformation matrix representation:

$$T(q) = \begin{bmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{bmatrix}$$

For a given configuration q, if we apply a control (v_x, v_y, ω), we define the *homogeneous representation* of the rotation center of u at q to be $T(q)u$.

For a given control line $L = (k_x, k_y, k_\theta)$, there is a corresponding transformation matrix T_L that transform a configuration in the world frame to the *control line frame*:

$$T_L = \begin{bmatrix} k_x & k_y & 0 \\ k_y & k_x & k_\theta \\ 0 & 0 & 1 \end{bmatrix}$$

For a configuration q in the world frame, we use $q^L = T_L q$ to denote its representation in the control line frame whenever the control line L is clear from the text.

A nice property of the control line frame is that, for a given control line L, a given configuration q in the world frame, and a control u, we can compute the corresponding Hamiltonian value of applying u at q along the control line L as $H = (T_L T(q)u)^T \cdot (0, 1, 0)$. We will show how to construct the control line for a given Hamiltonian value in the next section. The Hamiltonian value of applying u at q along the control line L may be different from the Hamiltonian value used to construct the control line. If so, this shows that applying u at q will not satisfy necessary conditions for optimal trajectories.

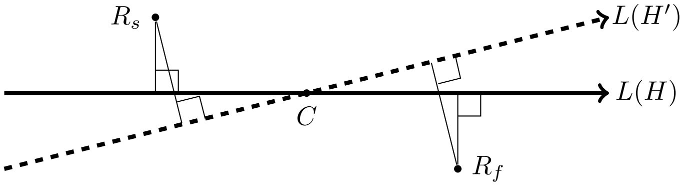


Figure 3.11: Illustration of proof of Theorem 3.6 in the case that $g \neq 0$. The rotation centers at q_s and q_f are denoted by R_s and R_f respectively. The construction of the control line is only based on algebra. The geometrical meaning is that the control line will pass through a point $C = (a', b')$ and for any rotation with angular velocity ω, the distance from the rotation center to the control line is H/ω.

In order to determine the duration of applying a control u before switch to another control u', it is convenient to define switch points. For two controls $u = (v_x, v_y, \omega)$ and $u' = (v'_x, v'_y, \omega')$, we define the *switch point* from u to u' be $p(u, u') = (v_y - v'_y, v'_x - v_x, \omega' - \omega)$.

3.11.2 Parametrization of the control lines

In this section, we show that when the start configuration q_s, the first control u_s, the final configuration q_f, the last control u_f, where one of u_s and u_f is a rotation, we can construct at most two mappings from the Hamiltonian values to control lines: $L_1(H)$ and $L_2(H)$. When both u_s and u_f are translations, the control line trajectories are TGT trajectories. Since we treat TGT trajectories separately, we do not deal with TGT trajectories in this section.

Theorem 3.6. For a given start configuration q_s, given goal configuration q_f, first control u_s, and the last control u_f, we can determine an interval of Hamiltonian values $I = (0, H_u)$, such that there exists mappings, $L_1(H)$ and $L_2(H)$, from Hamiltonian values in I to control lines.

Proof. Since $k_x^2+k_y^2 = 1$, we can represent the control line as $(k_x, k_y, k_\theta) = (\cos\varphi, \sin\varphi, k_\theta)$. Let $R_s = (a_s, b_s, g_s)$ be the homogeneous representation of the rotation center of u_s at q_s. Let $R_f = (a_f, b_f, g_f)$ be the homogeneous representation of the rotation center of u_f at q_f. Let $a = a_s - a_f$, $b = b_s - b_f$, and $g = g_s - g_f$. By Equation 3.3, we have:

$$-k_y a_s + k_x b_s + k_\theta g_s = H \tag{3.9}$$

$$-k_y a_f + k_x b_f + k_\theta g_f = H \tag{3.10}$$

$$-k_y a + k_x b + k_\theta g = 0 \tag{3.11}$$

There are two cases: $u_{s,\omega} \neq u_{f,\omega}$ and $u_{s,\omega} = u_{f,\omega}$.

The case of $u_{s,\omega} \neq u_{f,\omega}$ ($g \neq 0$).

By Equation 3.11, we have $k_\theta = (k_y a - k_x b)/g$. By substitute back in Equation 3.10, we have

$$-k_y a_f + k_x b_f + (k_y a - k_x b) g_f / g = H \tag{3.12}$$

Combing with Equation 3.9, we have,

$$-k_y(a_f - (a g_f)/g) + k_x(b_f - (b g_f)/g) = H \tag{3.13}$$

By setting $a' = a_f - (a g_f)/g$ and $b' = b_f - (b g_f)/g$, we express Equation 3.13 differently as follows.

$$-a' \sin\varphi + b' \cos\varphi = H \tag{3.14}$$

The geometrical meaning is that the control line will pass through a point $C = (a', b')$ and for any rotation with angular velocity ω, the distance from the rotation center to the control line is H/ω.

Let $r^2 = a'^2 + b'^2$, $\alpha = \text{atan}(a', b')$. We have $a' = r \sin \alpha$ and $b' = r \cos \alpha$. By trigonometric identities, we have

$$r \cos(\varphi + \alpha) = H$$

Since $\cos(\varphi + \alpha) = H/r$, $\sin(\varphi + \alpha) = \pm\sqrt{1 - H^2/r^2} = \pm\sqrt{r^2 - H^2}/r$.

$$k_x = \cos\varphi = \cos(\varphi+\alpha-\alpha) = \cos(\varphi+\alpha)\cos(\alpha)+\sin(\varphi+\alpha)\sin\alpha = \frac{b'H \pm a'\sqrt{r^2 - H^2}}{r^2}$$

$$k_y = \sin\varphi = \sin(\varphi+\alpha-\alpha) = \sin(\varphi+\alpha)\cos(\alpha)-\cos(\varphi+\alpha)\sin\alpha = \frac{\pm b'\sqrt{r^2 - H^2} - a'H}{r^2}$$

$$k_\theta = \frac{H + a_f k_y - b_f k_x}{g_f} = \frac{ak_y - bk_x}{g}$$

Thus, we have two control lines:

$$L_1(H) = (\frac{b'H + a'\sqrt{r^2 - H^2}}{r^2}, \frac{b'\sqrt{r^2 - H^2} - a'H}{r^2}, \frac{ak_y - bk_x}{g}) \text{ and}$$
$$L_2(H) = (\frac{b'H - a'\sqrt{r^2 - H^2}}{r^2}, \frac{-b'\sqrt{r^2 - H^2} - a'H}{r^2}, \frac{ak_y - bk_x}{g}).$$

The case of $u_{s,\omega} = u_{f,\omega}$ ($g = 0$).

Let $r = a^2 + b^2$, $\alpha = \text{atan}(a, b)$. We have

$$r \cos(\varphi + \alpha) = 0$$

Since $\cos(\varphi + \alpha) = 0$, then $\sin(\varphi + \alpha) = \pm 1$.

$$k_x = \cos\varphi = \cos(\varphi + \alpha - \alpha) = \cos(\varphi + \alpha)\cos(\alpha) + \sin(\varphi + \alpha)\sin\alpha = \pm a/r$$

$$k_y = \sin\varphi = \sin(\varphi + \alpha - \alpha) = \sin(\varphi + \alpha)\cos(\alpha) - \cos(\varphi + \alpha)\sin\alpha = \pm b/r$$

$$k_\theta = \frac{H + a_f k_y - b_f k_x}{g_f}$$

Thus, we have two control lines

$$L_1(H) = (a/r, b/r, \frac{H + a_f k_y - b_f k_x}{g_f}) \text{ and } L_2(H) = (-a/r, -b/r, \frac{H + a_f k_y - b_f k_x}{g_f}).$$

For a given control set, there exists an upper bound H_u of the Hamiltonian values that will correspond to non-trivial control line trajectories. Thus, we limit the range of $L_1(H)$ and $L_2(H)$ to be smaller than H_u. □

The parametrization we present here has different form from the parametrization under cost-free-switch model (Furtuna 2011), which is based on trigonometric functions. Our parametrization is easier to analyze, but Furtuna's parametrization is more numerically stable, and hence we use Furtuna's parametrization in the implementation.

3.11.3 Durations of controls

In this section, we show that given a control line L, a configuration q, and two different controls u and u', the duration of applying u at a configuration q until switching to control u' can have at most two possibilities.

Theorem 3.7. Given a control line L and a non-singular interval, the duration of applying a control u from a configuration q until switching to another control u' has at most two possibilities such that the resulting motion can be a subtrajectory of a control line trajectory corresponding to the control line L. Moreover, these two possibilities can be determined analytically.

Proof. Suppose that a control line trajectory applies u at $q^L = T_L q$ and then switches to control u' at a configuration $\hat{q}^L$ in the control line frame. By BIP, at the moment of switching control at configuration $\hat{q}^L$, the Hamiltonian values for u and u' must be the same. Based on Furtuna's analysis (Furtuna 2011), this implies $(T(\hat{q}^L)p(u,u'))^T \cdot [0,1,0] = 0$. That is, when we attach $p(u,u')$ to the robot, $p(u,u')$ will lie on the control line when the robot is at $\hat{q}^L$. Hence, we can solve for $\hat{q}^L$ as follows:

Let $p^L = T_L T(q) p(u,u') = (x_p, y_p, w_p)$ be the switching point in the control line frame when the robot is at q in the world frame. Let $r^L = T_L T(q) u = (x_r, y_r, \omega)$ be the homogeneous representation of the rotation center of u at q^L in the control line frame. If u is a translation and $w_p \neq 0$, then the duration $t = y_p / x_r$. If u is a rotation, then t must satisfy

$$b_1 \sin(\omega t) + b_2 \cos(\omega t) + b_3 = 0$$

where

$$b_1 = x_p - w_p x_r / \omega$$

$$b_2 = y_p - w_p y_r / \omega$$

$$b_3 = w_p y_r / \omega$$

The solution of Equation 3.11.3 is

$$\omega t = \operatorname{atan}(b_1 \pm \sqrt{b_1^2 + b_2^2 - b_3^2}, b_2 - b_3) + 2\pi n, \forall n \in \mathcal{Z}$$

Since we are only interested in the solution with $t > 0$ and $|\omega t| < 2\pi$, there are at most two solutions. □

3.11.4 Durations for a sequence of controls

By the method described above, for a sequence of controls $\mathbf{u}$, $|\mathbf{u}| = n$, we can determine all possible durations $\mathbf{t}$, each t_i, $1 \leq i < n$, has at most two solutions, but the duration of the last control is still undetermined. When we are given the last configuration q_f, we can determine the duration of the last control as follows: If u_f is a rotation, we apply u_f until the configuration has the same orientation as q_f. If u_f is a translation, then we consider the control u_{n-1}. There are only two possible configuration of switch from u_{n-1} to u_f, such that applying u_f will reach q_f, and each configuration has different durations for u_f. Thus, there are two possible durations of u_f and we can choose the one which is closer to q_f.

Consequently, given a control line L and a sequence of control $\mathbf{u}$, $|\mathbf{u}| = n$, there are at most 2^{n-1} possible durations $\mathbf{t}$ so that the trajectory $(\mathbf{u}, \mathbf{t})$ corresponds to the control line L. Furthermore, if we fix the way to determine the duration according to *selectors* $b_1, \ldots, b_{n-1}$, $b_i \in \{1, 2\}$ so that the duration t_i is the b_i-th solution, then the duration is fully determined by the control line L. Thus, we have the following theorem.

Theorem 3.9. Let $\mathbf{u} \in U^n$ be a sequence of controls. Let $\mathbf{b} \in \{1, 2\}^{n-1}$ be a sequence of *selectors*. For a given control line L with a non-critical Hamiltonian value H, start configuration q_s, and final configuration q_f, the duration of each control is fully determined by $\mathbf{u}$ and $\mathbf{b}$.

3.11.5 Finiteness of trajectories corresponding to a control line L

Since we have an upper bound for B for any optimal trajectories, we can limit ourselves to at most $|U|^B$ possible control sequences for optimal trajectories. When a control line L and a fixed control sequence $\mathbf{u}$ are given, since the duration for each control can be determined with at most two possibilities, there are at most 2^B possible corresponding durations for this control sequence with respect to the control line L. Thus, we can limit ourselves to these at most $(2|U|)^B$ possible trajectories corresponding to the control line L.

3.12 Lipschitz continuity

Fix a pair of controls (u_s, u_f), an interval of Hamiltonian values I constructed in Theorem 3.10, and a mapping $L(H)$ from the Hamiltonian values to control lines, we show that the distance function computed in Section 3.10.3 is Lipschitz continuous with respect to the change of the Hamiltonian value H.

First consider the cost function c, which is the summation of the switching cost and the time cost of the trajectory. Consider a fixed sequence of controls $\mathbf{u}$ and we analyze the dependency of its duration $\mathbf{t}$ on the Hamiltonian value. By Theorem 3.9, when we fix a sequence of controls and selectors, the duration is fully determined by

$L(H)$. Since Lipschitz continuity is closed under the minimum operation, it suffices to prove that for any sequence of controls $\mathbf{u}$ and selectors, the cost is Lipschitz continuous with respect to the change of the Hamiltonian value $H \in I$. Since the sequence is unchanged, the switching cost stays the same for any Hamiltonian value $H \in I$. Thus, it suffices to prove that for any sequence of controls $\mathbf{u}$ and selectors, the durations $\mathbf{t}$ is Lipschitz continuous with respect to the change of the Hamiltonian value $H \in I$. Similarly, we prove that for any sequence of controls $\mathbf{u}$ and selectors, the distance from the trajectory to the goal is Lipschitz continuous with respect to the change of the Hamiltonian value $H \in I$.

In order to simplify the analysis, in the remaining part of this section, we consider fixed selectors so that $\mathbf{t}$ is a function of the Hamiltonian value H directly without ambiguity.

3.12.1 Lipschitz continuity of $d(L(H), \mathbf{u}, \mathbf{t})$ and $c(L(H), \mathbf{u}, \mathbf{t})$

Let $(\mathbf{u}, \mathbf{t})$, $|\mathbf{u}| = n$ be a regular trajectory corresponding to the control line $L(H)$. We first consider the cost function $c(L(H), \mathbf{u}, \mathbf{t})$, which depends on the durations of each control and the switching cost. Since the sequence is unchanged, the switching cost will not change and hence we focus on durations. Let $t_i(H)$ be the duration for the i-th control u_i with respect to H. Since $c(L(H), \mathbf{u}, \mathbf{t})$ is the summation of all t_i, it suffices to prove that each $t_i(H)$ is Lipschitz continuous.

Second, we consider the distance function $d(L(H), \mathbf{u}, \mathbf{t})$. For control u_i and its corresponding sub-trajectory, we use d_i to denote the length of the sub-trajectory projection onto the control line. The distance function $d(L(H), \mathbf{u}, \mathbf{t})$ can be rewritten as $|q^L_{s,x} + \sum_{i=1}^{n} d_i - q^L_{f,x}|$. It suffices to show that each $d_i(H)$ and the mapping T_L is

Lipschitz continuous.

Durations $t_i(H)$ and projections $d_i(H)$, $1 < i < n$ are easier to analyze, since they depend on H directly. However, durations $t_1(H)$ and $t_n(H)$ depend on H, start configuration q_s^L, and final configuration q_f^L in the control line frame, which depends on H. Hence, $t_1(H)$ and $t_n(H)$ depend on H not only directly but also indirectly through q_s^L and q_f^L. Similarly, $d_1(H)$ and $d_n(H)$ also depend on H directly and indirectly. The analysis of $t_1(H)$, $t_n(H)$, $d_1(H)$, and $d_n(H)$ should be separated from the analysis of $t_i(H)$ and $d_i(H)$, $1 < i < n$. Thus, we analyze $t_i(H)$ and $d_i(H)$, $1 < i < n$ first and then show the mapping T_L is Lipschitz continuous. Finally, we give the analysis of $t_1(H)$, $t_n(H)$, $d_1(H)$, and $d_n(H)$.

3.12.2 Analysis of $t_i(H)$ and $d_i(H)$, $1 < i < n$

Theorem 3.12. *Let $I = (a, b)$ be an open interval of the partition of the Hamiltonian values. Let $\mathbf{u}$ be a fixed sequence of n controls. Let $t_i(H)$ be the duration for the u_i and $d_i(H)$ be the length of projection of the sub-trajectory corresponding to u_i onto the control line. For any $\delta > 0$, both functions $t_i(H)$ and $d_i(H)$ are Lipschitz continuous with respect to the Hamiltonian values $H \in (a, b - \delta)$ for all $1 < i < n$.*

Proof. The duration $t_i(H)$ and length $d_i(H)$ are fully determined by u_{i-1}, u_i, u_{i+1}, and H. Let q_i^L be the configuration in the control line frame at which the trajectory switches control from u_{i-1} to u_i. Let q_{i+1}^L be the configuration in the control line frame at which the trajectory switches control from u_i to u_{i+1}. Here, we use a result from Furtuna's Thesis (Furtuna 2011) that there exists a point $p_i = p(u_{i-1}, u_i)$ rigidly attached to the robot, such that p_i will lie on the control line when the robot is at

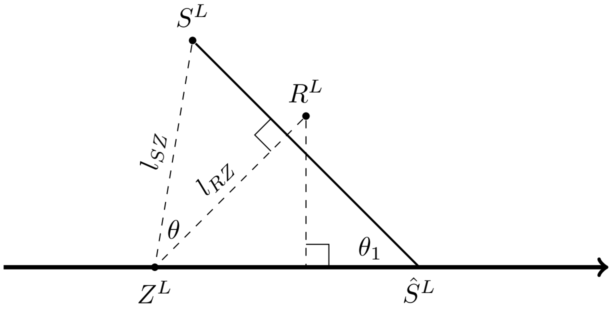


Figure 3.12: Illustration of proof of Theorem 3.11 in the case that u_i is a translation. Initially, the switch point is located at S^L. A control line trajectory must apply u_i until the switch point collides with the control line at $\hat{S}^L$.

q_i^L. Similarly, when the robot is at q_{i+1}^L and switches from u_i to u_{i+1}, there exists a point $p_{i+1} = p(u_i, u_{i+1})$ attached to the robot such that p_{i+1} is on the control line.

We introduce some notation for the remainder of the proof. Let $Z^L = (Z_x^L, 0)$ be the location of p_i attached to the robot at q_i^L, which is on the control line. Let $S^L = (S_x^L, S_y^L)$ be the location of p_{i+1} attached to the robot at q_i^L. Let $\hat{S}^L = (\hat{S}_x^L, 0)$ be the location of p_{i+1} attached to the robot at q_{i+1}^L. By considering the position of S^L we can determine the $t_i(H)$ and $d_i(H)$.

Depending on whether u_i is a translation or not, there are two cases:

The case in which u_i is a translation.

Let v_i be the velocity of u_i. By Theorem 3.1, the magnitude of the projection of the velocity onto the control line is H. Consequently, the magnitude of velocity in the y-coordinate in the control line frame is $v_y^L = \sqrt{v_i^2 - H^2}$. The duration of t_i can be computed as S_y^L / v_y^L. Consequently, the length of the projection of the trajectory onto the control line, $d_i(H)$, can be computed as $t_i H$. Hence, it suffices to prove t_i is Lipschitz continuous.

The control u_{i-1} must be a rotation, since if u_{i-1} is a translation, then u_i and u_{i-1} have the same Hamiltonian value along the sub-trajectory corresponding to u_i and the trajectory is a singular trajectory. Let $R^L = (R^L_x, R^L_y)$ be the location of the rotation center of control u_{i-1}. Let l_{SZ} be the distance between S^L and Z^L. Let l_{RZ} be the distance between R^L and Z^L. Let θ be the angle rotating from vector Z^LS^L to vector Z^LR^L counterclockwise. Since the the mutual distance among S^L, R^L and Z^L is independent from H, l_{RZ} and θ are independent from H.

Let θ_1 be the angle between segment $S^L\hat{S}^L$ and the control line; the value of θ_1 is $\operatorname{acos}(H/v_i)$. Furthermore, it can be shown that the line Z^LR^L is perpendicular to the line $S^L\hat{S}^L$ (Furtuna 2011). By geometric reasoning, S^L_y can be computed as $l_{SZ}\cos(\theta - \operatorname{acos}(H/v_i)) = (l_{SZ}/v_i)(H\cos\theta + \sqrt{v_i^2 - H^2}\sin\theta)$. Remember all the durations is capped by M, the cost of a feasible trajectory. Hence, we have

$$t_i(H) = \min(M, \frac{S^L_y}{v^L_y} = \frac{l_{SZ}}{v_i}\left(\sin\theta + \frac{H\cos\theta}{\sqrt{v_i^2 - H^2}}\right)).$$

Since the second term is monotonically increasing in H, there exists a threshold γ that for all $H \geq \gamma$, $t_i = M$. Thus, within the interval $[\gamma, b)$, $t_i(H)$ is a constant. Hence, we only focus on the part (a, γ) that the minimum is taken from the second term.

A differentiable function is Lipschitz continuous if this function has a bounded first derivative.

$$\frac{\partial t_i(H)}{\partial H} = \left(\frac{v_i l_{SZ}\cos\theta}{(v_i^2 - H^2)^{1.5}}\right)$$

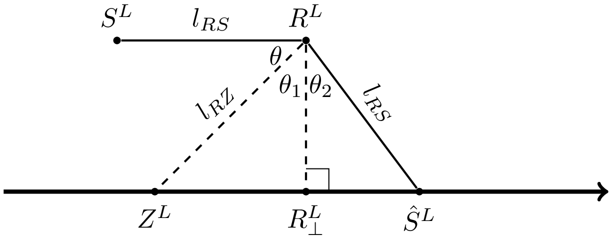


Figure 3.13: Illustration of proof of Theorem 3.11 in the case that u_i is a rotation, $\omega_{i-1} > \omega_i$, and $\omega_{i+1} > \omega_i$. Initially, the switch point is located at S^L. A control line trajectory must apply u_i, a rotation around R^L counterclockwisely, until the switch point collides with the control line at $\hat{S}^L$.

Since $d_i = t_i H$, we have

$$\frac{\partial d_i(H)}{\partial H} = H\frac{\partial t_i}{\partial H} + t_i = \left(\frac{v_i l_{SZ}\cos\theta}{(v_i^2 - H^2)^{1.5}}\right) + \frac{l_{SZ}}{v_i}\left(\sin\theta + \frac{H\cos\theta}{\sqrt{v_i^2 - H^2}}\right).$$

For all $H \in (a, \gamma)$ and $H < v_i$, the derivatives of $t_i(H)$ and $d_i(H)$ are bounded.

The case in which u_i is a rotation.

Let $R^L = (R^L_x, R^L_y)$ be the location of the rotation center of control u_i and let $R^L_\perp = (R^L_x, 0)$ be the projection of R^L on the control line. We want to compute the angle, φ_0, between the control line to the vector $R^L S^L$, and the angle φ_1, between the control line to the vector $R^L \hat{S}^L$; these angles are measured in counterclockwise direction. The duration $t_i(H)$ can be computed as $(\varphi_1 - \varphi_0)/\omega_i$, where the subtraction wrapping around 2π and the result has the same sign as ω_i. Let r be the distance between the reference point of the robot and R^L when robot is at q^L_i. The projection of the trajectory on the control line, $d_i(H)$, can be computed as $r(\cos\varphi_1 - \cos\varphi_0)$. Thus, it suffices to show that φ_0 and φ_1 are Lipschitz continuous with respect to H.

Let l_{RZ} be the distance between R^L and Z^L and let l_{RS} be the distance between

R^L and S^L. Let θ be the angle rotating from vector R^LZ^L to R^LS^L counterclockwise. Note that θ, l_{RZ}, and l_{RS} are independent from H.

Let θ_1 be the angle between the segment R^LZ^L and $R^LR^L_\perp$, which equals $\operatorname{acos}(H/(l_{RZ}\omega_i))$. Let θ_2 be the angle between the segment $R^L\hat{S}^L$ and $R^LR^L_\perp$, which equals $\operatorname{acos}(H/(l_{RS}\omega_i))$. Let ω_{i-1} and ω_{i+1} be the angular velocity of u_{i-1} and u_{i+1} respectively. Based on θ_1 and θ_2, we can compute φ_0 and φ_1 as follows:

		φ_0		φ_1
$\omega_i > 0$	$Z^L_x \leq R^L_x$	$3\pi/2 - \theta_1 + \theta$	$\hat{S}^L_x \geq R^L_x$	$3\pi/2 + \theta_2$
$\omega_i > 0$	$Z^L_x > R^L_x$	$3\pi/2 + \theta_1 + \theta$	$\hat{S}^L_x < R^L_x$	$3\pi/2 - \theta_2$
$\omega_i < 0$	$Z^L_x > R^L_x$	$\pi/2 - \theta_1 + \theta$	$\hat{S}^L_x < R^L_x$	$\pi/2 + \theta_2$
$\omega_i < 0$	$Z^L_x \leq R^L_x$	$\pi/2 + \theta_1 + \theta$	$\hat{S}^L_x \geq R^L_x$	$\pi/2 - \theta_2$

Thus, we have

$$\left|\frac{\partial\varphi_0(H)}{\partial H}\right| \leq ((l_{RZ}\omega_i)^2 - H^2)^{-0.5} \text{ and } \left|\frac{\partial\varphi_1(H)}{\partial H}\right| \leq ((l_{RS}\omega_i)^2 - H^2)^{-0.5}.$$

Consequently,

$$\left|\frac{\partial t_i(H)}{\partial H}\right| \leq \frac{((l_{RZ}\omega_i)^2 - H^2)^{-0.5} + ((l_{RS}\omega_i)^2 - H^2)^{-0.5}}{|\omega_i|}.$$

$$\begin{aligned}\left|\frac{\partial d_i(H)}{\partial H}\right| \leq \frac{r}{l_{RZ}|\omega_i|}&\left(|\sin\theta_1| + \left|\frac{H\cos\theta_1}{\sqrt{(l_{RZ}\omega_i)^2 - H^2}}\right|\right)\\ &+ \frac{r}{l_{RS}|\omega_i|}\left(|\sin\theta_2| + \left|\frac{H\cos\theta_2}{\sqrt{(l_{RS}\omega_i)^2 - H^2}}\right|\right)\end{aligned}$$

By the construction of the partition of the Hamiltonian values, $b \leq |l_{RZ}\omega_i|$ and $b \leq |l_{RS}\omega_i|$ so that the switch of controls is feasible. For any $\delta > 0$, when $H \in (a, b-\delta)$,

H is smaller than $|l_{RZ}\omega_i|$ and $|l_{RS}\omega_i|$, and the derivatives of $t_i(H)$ and $d_i(H)$ are bounded. □

During the analysis, we also fully analyze the duration for three consecutive controls u, u', and u'' in a control line trajectory with respect to a given H value.

Theorem 3.8. Let u, u', and u'' be three consecutive controls in a control line trajectory. Given k_x, k_y, and a non-critical Hamiltonian value H, the duration of u' has at most two possibilities and can be determined analytically without knowing the configuration.

Before we analyze d_1, t_1, d_n, and t_n, since they depend on the mapping $T_L(q) = q^L$, they indirectly depend on k_x, k_y, and k_θ. Hence, we show the analysis of k_x, k_y, k_θ first and then analyze q^L before we analyze d_1, t_1, d_n, and t_n.

3.12.3 Analysis of k_x, k_y, k_θ, and $\text{atan}(k_y, k_x)$

Since the mapping T_L depends on k_x, k_y, and k_θ, we analyze the dependency of k_x, k_y, and k_θ on the Hamiltonian value H. Furthermore, we also analyze $\text{atan}(k_y, k_x)$, since we will need it in the following sections.

Theorem 3.13. *For a mapping $L(H) = (k_x(H), k_y(H), k_\theta(H))$ from the Hamiltonian values to control lines with a domain $(0, H_u)$. The mapping $L(H)$ and $\text{atan}(k_y(H), k_x(H))$ are Lipschitz continuous with respect to the Hamiltonian values in $(0, H_u - \delta)$ for any $\delta > 0$.*

Proof. Since $k_\theta(H) = \frac{H + a_f k_y - b_f k_x}{g_f}$, we have

$$\frac{\partial k_\theta(H)}{\partial H} = \frac{1 + a_f \frac{\partial k_y(H)}{\partial H} - b_f \frac{\partial k_x(H)}{\partial H}}{g_f}$$

3.12 Lipschitz continuity

By using the notation in Section 3.11.2, there are two cases.

The case of $u_{s,\omega} = u_{f,\omega}$ $(g = 0)$

In this case, k_x and k_y will not change with respect to H. Hence, $\frac{\partial k_x(H)}{\partial H} = \frac{\partial k_y(H)}{\partial H} = \frac{\partial \mathrm{atan}(k_y, k_x)}{\partial H} = 0$ and $\frac{\partial k_\theta(H)}{\partial H} = 1/g_f$.

The case of $u_{s,\omega} \neq u_{f,\omega}$ $(g \neq 0)$

$$\frac{\partial k_x(H)}{\partial H} \leq \frac{|b'| + |\frac{a'H}{\sqrt{r^2 - H^2}}|}{r^2}$$

$$\frac{\partial k_y(H)}{\partial H} \leq \frac{|a'| + |\frac{b'H}{\sqrt{r^2 - H^2}}|}{r^2}$$

$$
\begin{aligned}
&\frac{\partial \text{atan}(k_y(H), k_x(H))}{\partial H} \\
&= \frac{1}{1 + k_y^2/k_x^2} \frac{\partial \frac{k_y(H)}{k_x(H)}}{\partial H} \\
&= k_x^2 \frac{\frac{\partial k_x(H)}{\partial H} k_y - k_x \frac{\partial k_y(H)}{\partial H}}{k_x^2} \\
&= \frac{\partial k_x(H)}{\partial H} k_y - k_x \frac{\partial k_y(H)}{\partial H} \\
&= \frac{b' + \frac{a'H}{\sqrt{r^2-H^2}}}{r^2} \frac{b'\sqrt{r^2-H^2} - a'H}{r^2} - \frac{b'H + a'\sqrt{r^2-H^2}}{r^2} \frac{-a' + \frac{b'H}{\sqrt{r^2-H^2}}}{r^2} \\
&= \frac{1}{r^4} \left((b' + \frac{a'H}{\sqrt{r^2-H^2}})(b'\sqrt{r^2-H^2} - a'H) - (b'H + a'\sqrt{r^2-H^2})(-a' + \frac{b'H}{\sqrt{r^2-H^2}})) \right) \\
&= \frac{1}{r^4} \left(b'^2\sqrt{r^2-H^2} - a'b'H + a'b'H - \frac{a'^2H^2}{\sqrt{r^2-H^2}} + a'b'H - \frac{b'^2H^2}{\sqrt{r^2-H^2}} + a'^2\sqrt{r^2-H^2} - a'b'H \right) \\
&= \frac{1}{r^4} \left((a'^2 + b'^2)(\sqrt{r^2-H^2}) - \frac{(a'^2+b'^2)H^2}{\sqrt{r^2-H^2}} \right) \\
&= \frac{1}{r^4}(a'^2 + b'^2)(\sqrt{r^2-H^2} - \frac{H^2}{\sqrt{r^2-H^2}}) \\
&= \frac{r^2 - 2H^2}{r^2\sqrt{r^2-H^2}} \\
&\leq \frac{1}{\sqrt{r^2-H^2}}
\end{aligned}
$$

For any $\delta > 0$, since $r \geq H_u$, when $H \in (0, H_u - \delta)$, we have $H < r$. Thus, for any $\delta > 0$, $k_x(H)$, $k_y(H)$, $k_\theta(H)$, and $\text{atan}(k_y(H), k_x(H))$ are Lipschitz continuous with respect to the Hamiltonian value $H \in (0, H_u - \delta)$. □

3.12.4 Analysis of q^L

For different Hamiltonian values, the mapping from q_s and q_f to the control line frame may be different. However, we can show that q_L is Lipschitz continuous with respect to the Hamiltonian value H.

Theorem 3.14. *For a mapping $L(H) = (k_x(H), k_y(H), k_\theta(H))$ from the Hamiltonian values to control lines with a domain $(0, H_u)$. The control line transformation $T_L(q)$ is Lipschitz continuous with respect to the Hamiltonian values in $(0, H_u - \delta)$ for any $\delta > 0$.*

Proof. For a configuration $q = (x, y, \cos\theta, \sin\theta)$ and a control line parametrized by H, $(k_x(H), k_y(H), k_\theta(H))$, the mapping from q to the control line frame is

$$q_L(H) = \begin{pmatrix} k_x(H)x + k_y(H)y \\ -k_y(H)x + k_x(H)y + k_\theta \\ k_x(H)\cos\theta - k_y(H)\sin\theta \\ k_x(H)\sin\theta + k_y(H)\cos\theta \end{pmatrix}$$

Hence,

$$\frac{\partial q_L(H)}{\partial H} = \begin{pmatrix} \frac{\partial k_x(H)}{\partial H}x + \frac{\partial k_y(H)}{\partial H}y \\ -\frac{\partial k_y(H)}{\partial H}x + \frac{\partial k_x(H)}{\partial H}y + \frac{\partial k_\theta(H)}{\partial H} \\ \frac{\partial k_x(H)}{\partial H}\cos\theta - \frac{\partial k_y(H)}{\partial H}\sin\theta \\ \frac{\partial k_x(H)}{\partial H}\sin\theta + \frac{\partial k_y(H)}{\partial H}\cos\theta \end{pmatrix}$$

For any $\delta > 0$, since $k_x(H)$, $k_y(H)$, and $k_\theta(H)$ are Lipschitz continuous with respect to the Hamiltonian value $H \in (0, H_u - \delta)$, q^L is Lipschitz continuous with respect to the Hamiltonian value $H \in (0, H_u - \delta)$ as well. □

3.12.5 Analysis of d_1, t_1, d_n, and t_n

In this section, we show that d_1, t_1, d_n, and t_n is Lipschitz continuous with respect to the change of the Hamiltonian value H.

Theorem 3.15. *Let $I = (a, b)$ be an open interval of the partition of the Hamiltonian values. Let $\mathbf{u}$ be a fixed sequence of n controls. Let $t_i(H)$ be the duration for the u_i and $d_i(H)$ be the length of projection of the sub-trajectory corresponding to u_i onto the control line. For any $\delta > 0$, d_1, t_1, d_n, and t_n are Lipschitz continuous with respect to the Hamiltonian values $H \in (a, b - \delta)$.*

Proof. Let $(0, H_u)$ be the domain of the mapping $L(H) = (k_x(H), k_y(H), k_\theta(H))$. For any $\delta > 0$, since $b \leq H_u$, k_x, k_y, k_θ, $\text{atan}(k_y, k_x)$, and q^L are Lipschitz continuous with respect to the Hamiltonian values in $(a, b - \delta)$ by Theorems 3.13 and 3.14

Since the cases for t_1 and t_n are symmetric, we only analyze t_1 and similarly we only analyze d_1. Let the start configuration be $q = (x, y, \cos\theta, \sin\theta)$ in the world frame. Fix a pair of controls (u, u'). Let $t(H)$ be the duration of applying u until switching to u'. Let $\hat{q}^L(H)$ be the configuration of the robot when the control switch from u to u'. Let $d(H) = \hat{q}^L_x - q^L_x$ be the projection of the motion on to the control line. There are two cases depending on whether u is a translation or a rotation.

The case in which u is a translation.

Let $S = (p_x, p_y)$ be the switch point of between u and u' in the world frame. Let $v = (v_x, v_y, 0)$ be the velocity in the world frame.

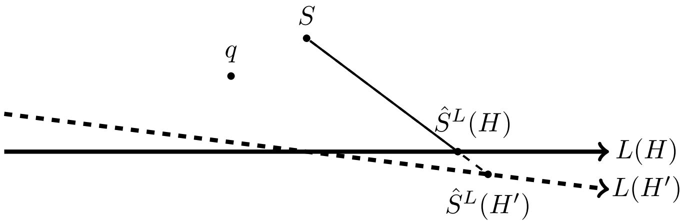


Figure 3.14: Illustration of proof of Theorem 3.15 in the case that u_i is a translation. The reference point of the robot is at q and the switch point is at p in the world frame. When the Hamiltonian value H changes, the control line changes from $L(H)$ to $L(H')$. Hence, the endpoint of the translation changes from $\hat{S}^L(H)$ to $\hat{S}^L(H')$ accordingly.

The representation of S in the control line frame S^L is

$$S^L(H) = \begin{pmatrix} k_x(H)p_x + k_y(H)p_y \\ -k_y(H)p_x + k_x(H)p_y + k_\theta(H) \end{pmatrix}$$

We have

$$\frac{\partial S^L_y(H)}{\partial H} = -\frac{\partial k_y(H)}{\partial H}p_x + \frac{\partial k_x(H)}{\partial H}p_y + \frac{\partial k_\theta(H)}{\partial H}$$

The velocity v_L in the control line frame is

$$\begin{aligned} v^L(H) &= \begin{bmatrix} k_x & k_y & 0 \\ -k_y & k_x & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & x \\ -\sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} k_x(\cos\theta v_x + \sin\theta v_y) + k_y(-\sin\theta v_x + \cos\theta v_y) \\ -k_y(\cos\theta v_x + \sin\theta v_y) + k_x(-\sin\theta v_x + \cos\theta v_y) \\ 0 \end{bmatrix} \end{aligned}$$

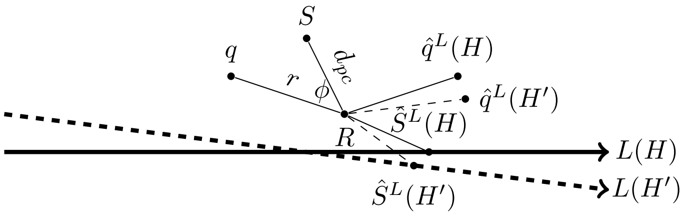


Figure 3.15: Illustration of proof of Theorem 3.15 in the case that u_i is a rotation. The reference point of the robot is at q and the switch point is at p in the world frame. When the Hamiltonian value H changes, the control line changes from $L(H)$ to $L(H')$. Hence, the endpoint of the rotation changes from $\hat{p}(H)$ to $\hat{p}(H')$ accordingly.

Let $v' = (v'_x, v'_y) = (\cos\theta v_x + \sin\theta v_y, -\sin\theta v_x + \cos\theta v_y)$. We have

$$\frac{\partial v_x^L(H)}{\partial H} = \frac{\partial k_x(H)}{\partial H} v'_x + \frac{\partial k_y(H)}{\partial H} v'_y \text{ and } \frac{\partial v_y^L}{\partial H} = -\frac{\partial k_y(H)}{\partial H} v'_x + \frac{\partial k_x(H)}{\partial H} v'_y$$

The time is

$$t(H) = -S_y^L / v_y^L = \frac{-k_y(H)p_x + k_x(H)p_y + k_\theta(H)}{-k_y(H)(\cos\theta v_x + \sin\theta v_y) + k_x(H)(-\sin\theta v_x + \cos\theta v_y)}$$

Since $\hat{q}_x^L(H) = q_x^L(H) + v_x^L t(H)$, $d(H) = q_x^L(H) + v_x^L t(H) - q_x^L(H) = v_x^L t(H)$. Consequently,

$$\frac{\partial d(H)}{\partial H} = \frac{\partial v_x^L(H)}{\partial H} t(H) + v_x^L(H) \frac{\partial t(H)}{\partial H}$$

For any $\delta > 0$, since v_y^L and S_y^L are Lipschitz continuous with respect to the Hamiltonian values in $(a, b - \delta)$, $d(H)$ and $t(H)$ are Lipschitz continuous with respect to the Hamiltonian values in $(a, b - \delta)$ as well.

The case in which u is a rotation.

Let $S = (p_x, p_y)$ be the switch point of between u and u' in the world frame. Let R be the rotation center of u in the world frame. Let r be the radius of rotation. Let

d_{pc} be the distance between p and c. Let φ be the angle between the vector from c to p and the vector from c to q. Note that S, R, r, d_{pc}, q, and φ are independent from the change of the Hamiltonian value.

Let $\hat{S}^L(H)$ be the position of the switch point when the robot is at $\hat{q}^L(H)$, where $\hat{S}^L(H)$ should be on the control line. Let $\alpha(H)$ be the angle between the vector from R to $\hat{S}^L(H)$ and the control line.

$$\sin\alpha = \frac{H}{\omega d_{pc}}, \cos\alpha = \pm\frac{\sqrt{\omega^2 d_{pc}^2 - H^2}}{\omega d_{pc}}$$

The time $t(H) = (\varphi(H) - \alpha(H))/\omega$. Thus,

$$\begin{aligned}
|\frac{\partial t(H)}{\partial H}| &\leq \left(|\frac{\partial \mathrm{atan}(k_y, k_x)(H)}{\partial H}| + \frac{1}{\sqrt{1 - (H^2)/(\omega d_{pc})^2}}\right)/|\omega| \\
&= \left(|\frac{\partial \mathrm{atan}(k_y, k_x)(H)}{\partial H}| + \frac{|\omega| d_{pc}}{\sqrt{(\omega d_{pc})^2 - H^2}}\right)/|\omega|
\end{aligned}$$

By the construction of the partition of Hamiltonian values, $b \leq \omega d_{pc}$ so that the switch of controls is feasible. For any $\delta > 0$, when $H \in (a, b-\delta)$, H is smaller than ωd_{pc}. Since $\mathrm{atan}(k_y, k_x)$ is Lipschitz continuous with respect to the Hamiltonian values in $(a, b-\delta)$, $d(H)$ is Lipschitz continuous with respect to the Hamiltonian values in $(a, b-\delta)$ as well.

Remember that $d(H) = q_x^L - \hat{q}_x^L$. We analyze $\hat{q}_x^L$ first.

$$\begin{aligned}
\hat{q}_x^L - R_x^L &= r\cos(\varphi + \alpha) \\
&= r(\cos\varphi\cos\alpha - \sin\varphi\sin\alpha) \\
&= r\left(\cos\varphi\frac{\sqrt{\omega^2 d_{pc}^2 - H^2}}{\omega d_{pc}} - \sin\varphi\frac{H}{\omega d_{pc}}\right) \\
&= \frac{r}{\omega d_{pc}}(\cos\varphi\sqrt{\omega^2 d_{pc}^2 - H^2} - \sin\varphi H)
\end{aligned}$$

Thus,

$$|\frac{\partial \hat{q}_x^L(H)}{\partial H}| \le |\frac{\partial R_x^L(H)}{\partial H}| + \frac{r}{|\omega| d_{pc}}(|\sin\varphi| + |\cos\varphi|\frac{H}{\sqrt{(\omega d_{pc})^2 - H^2}})$$

Hence,

$$|\frac{\partial d(H)}{\partial H}| \le |\frac{\partial q_x^L}{\partial H}| + |\frac{\partial R_x^L}{\partial H}| + \frac{r}{|\omega| d_{pc}}(|\sin\varphi| + |\cos\varphi|\frac{H}{\sqrt{\omega^2 d_{pc}^2 - H^2}})$$

For any $\delta > 0$, since $q_x^L(H)$ and $R_x^L(H)$ are Lipschitz continuous with respect to the Hamiltonian values in $(a, b - \delta)$, $d(H)$ is Lipschitz continuous with respect to the Hamiltonian values in $(a, b - \delta)$ as well. □

By Theorem 3.12 and 3.15, we have the following theorem.

Theorem 3.11. Let $I = (a, b)$ be an open interval of the partition of the Hamiltonian values. Let $\mathbf{u}$ be a fixed sequence of n controls. Let $t_i(H)$ be the duration for the u_i and $d_i(H)$ be the length of projection of the sub-trajectory corresponding to u_i onto the control line. For any $\delta > 0$, both functions $t_i(H)$ and $d_i(H)$ are Lipschitz continuous with respect to the Hamiltonian values $H \in (a, b - \delta)$ for all $1 \le i \le n$.

3.13 Implementation

We implemented the algorithm in C++. Our testing environment is a desktop system with an Intel Xeon W3550 3.07 GHz CPU.

For the costly-switch model, we used three test cases. First, we used the bench mover's problem (Lyu et al. 2014) as one test case. We compared our program's result with the results of the analytical solver. Except for some cases in which the Hamiltonian value is close to the upper bound (for which numerical instability becomes a problem), the results coincide with the results from the exact solver.

We used the refrigerator-mover's problem as the second test case. The refrigerator-mover's problem is an extension of bench mover's problem, inspired by a problem in *Mechanics of Robotic Manipulation* (Mason 2001): a mover wants to move a refrigerator from one location and orientation to another. The refrigerator is too heavy to move by lifting or pushing, but it can be lifted onto any of the four legs at the corners of the square base and rotated. One approximately optimal trajectory is shown in Figure 3.10. Third, we used omni-directional vehicle as a test case; one approximately optimal trajectory is show in Figure 3.2b.

The solver described can also be used as a general-purpose solver for finding time-optimal trajectories of a kinematic planar rigid body under the cost-free-switch model. In this case, we additionally constrain the structure of the trajectories using the maximization condition of PMP, and apply the Lipschitz optimizer to find best trajectories for each possible structure. We applied this approach to the problem of finding optimal trajectories for the omnidirectional robot (Wang and Balkcom 2012a), and found that the approach was only about one order of magnitude slower (on the order of 0.03 seconds per configuration) than the special-purpose analytical solver (Wang and Balkcom 2012a). One approximately optimal trajectory is show in Figure 3.2a.

3.14 Conclusion and future work

By adding a cost for switching between controls, we ensure existence of solutions for optimal control problems, and evade the problem of chattering. By applying Blatt's Indifference Principle and Lipschitzian optimization approach, we can find approximately optimal trajectories, and the error can be forced to be arbitrarily small.

The implemented approach does have some limitations, and these limitations do suggest rich problems for future study. One of the limitations is that when applying Lipschitzian optimization techniques, the algorithm reduces the search domain by a user controlled parameter in order to make the optimization problem behave smoothly. Although the controlled parameter can set to any arbitrarily small number, this algorithm may not find optimal trajectories for some scenarios if the controlled parameter is not small enough. We believe this that issue can be resolved by reformulating the Lipschitzian optimization problem with other parametrization.

Moreover, the potential number of optimal trajectory structures can be very very large under the costly-switch model. For the costly-switch model, an algorithm might potentially need to explore a number of structures that is exponential in the number of controls in order to find solutions. For example, in order to find approximately optimal trajectories for omni-directional vehicle, whose control set contains fourteen controls, it takes about an hour to find an high-precision approximately optimal trajectory for a start configuration and goal configuration. We believe that better

Lipschitz constants, use of the derivative of the objective function together with more sophisticated approaches to Lipschitzian optimization (Lera and Sergeyev 2013), and a more directed A* search could dramatically reduce these costs.

Finally, in this work, we assume that there are no obstacles. Under the cost-free-switch model, Pontryagin's Maximum Principle can be extended to systems with state constraints (Maurer 1977; Maurer and Osmolovskii 2013). However, to the best of our knowledge, extending Blatt's Indifference Principle to systems with state constraints is still an unsolved task. Extending Blatt's Indifference Principle to systems with state constraints (or making use of the KKT approach) will be an interesting future direction of research.

Chapter 4

Multi-query for near-optimal trajectories

4.1 Introduction

This chapter examines how an *optimal steering method*, which finds optimal trajectories if there are no obstacles, may be used to build a good approximate cell-based representation of a configuration space with obstacles. We also present algorithms that make use of this representation to find good, although not necessarily optimal, paths.

Approaches to finding optimal paths among obstacles can be loosely classified into two types: cell decomposition (for example, early work by Barraquand and Latombe (Barraquand and Latombe 1993) and Xavier and Donald (Donald and Xavier 1995)), and sampling methods, such as PRM* (Karaman and Frazzoli 2011). Cell-based methods have the advantage that it can be easier to prove results about path quality after finite computation time, but typically require division of the space into a number

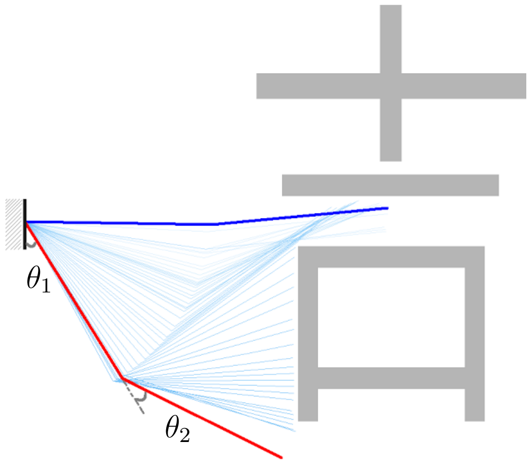


Figure 4.1: Initial and final positions are represented by thick lines, while others are intermediate positions of the arm.

of cells that is explicitly exponential in the dimensionality of the space. Sampling methods, on the other hand, can quickly return reasonable paths if there are large open spaces in the environment, even in high dimensions.

This work begins to explore the idea that some easy spaces (with large open regions, or large obstacle regions) can be represented easily, without giving up optimality, using a variable-sized cell decomposition approach. Figures 4.1 and 4.2 show an example of such a decomposition for the configuration space for a serial planar arm with two revolute joints, as well as a path obtained by a simple A* search across points on the boundaries of the cells. Note that because optimal trajectories are not unique for the chosen metric (Section 4.3.1) some “wiggle" can occur even in optimal trajectories.

Variable-sized cell-decomposition methods are hardly new. There are already some motion planners based on cell-decomposition (Wang, Chiang, and Yap 2015). The

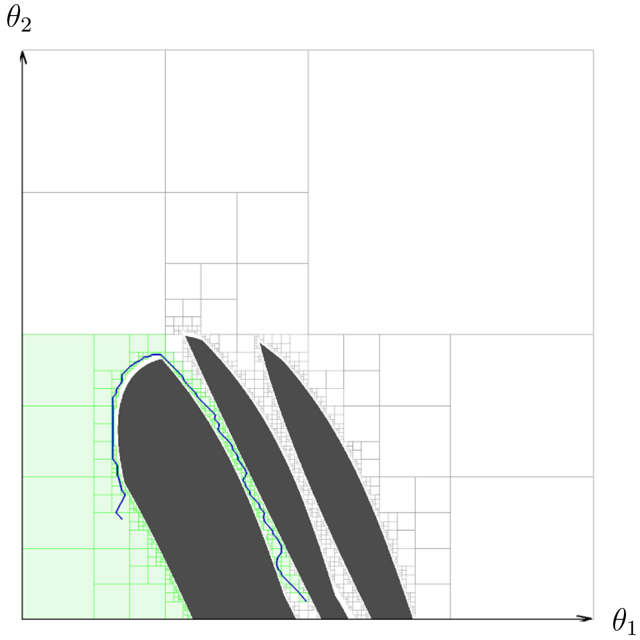


Figure 4.2: Decomposition of C-space and the corresponding trajectory in C-space. Shaded cubes are explored by the search algorithm.

unique contribution of this work is an exploration of the implications of optimality under some metric, for systems with and without non-holonomic constraints.

The key idea is to exploit convexity of regions in configuration space. Intuitively, between any pair of boundary points of a convex region, there should exist a path that does not leave the region. So, paths within the region need not be sampled or stored.

In this chapter, we show that a form of convexity can be defined (Section 4.2) that is suitable for configuration spaces with optimal steering methods and corresponding shortest-path metrics, and that a variant on this form of convexity, subconvexity, is available locally everywhere there is a reachable ball in the free configuration space. We consider some example systems, including a planar arm and a Reeds-Shepp car (Reeds and Shepp 1990), and present an algorithm for decomposing the free configuration space into cells (Section 4.4). We search this data structure for good paths (Section 4.6), and show some results for a few simple systems.

The work in this chapter is quite preliminary, but we believe that the theoretical framework hints at new ways to tackle issues such as ensuring good path quality after finite computation time (*vs.* weaker guarantees such as asymptotic optimality), topological analysis of configuration spaces, and compact representations of configuration spaces. Concretely, the current work suggests some bounds on how densely a space may need to be sampled to capture optimality over “most of" the configuration space. We believe that this is an important step to begin to understand the behavior of sampling-based planners near obstacles (a primary focus of, for example, variations such as obstacle-based PRM (Yeh et al. 2012) and Toggle PRM (Denny and Amato 2012)), after finite computation time.

4.1.1 Related work

We briefly summarize related work in kinodynamic motion planning and multi-query motion planning.

Kinodynamic motion planning

For a particle among obstacles in $\mathcal{R}^3$, shortest paths subject to kinodynamic constraints can be provably approximated (Donald et al. 1993; Reif and Tate 1994; Donald and Xavier 1995; Reif and Wang 2000); by growing the obstacles slightly, a lower bound on the required size of cells used to represent configuration space can be computed.

Recently, Bialkowski *et al.* have reduced the time cost of collision detection with

RRT*, by building balls in free (Euclidean) space in which collision detection needs to be performed only once (Bialkowski et al. 2012). Recent sampling-based algorithms by Li, Littlefield, and Bekris (Li, Littlefield, and Bekris 2016) achieve asymptotic optimality of path length in parameter space without the need for a steering method. Deits shows a numerical optimization approach to computing large convex regions, also in a Euclidean space (Deits and Tedrake 2014);

Multi-query motion planning

For multi-query motion planning, PRM was the first multi-query motion planner (Kavraki et al. 1996). One variant of PRM, PRM* (Karaman and Frazzoli 2011), is guaranteed to find optimal trajectories as the running time approaches infinity. However, in order to obtain optimal trajectories, the roadmap tends to have huge number of vertices and edges. Many motion planning algorithms that are guaranteed to find approximately optimal trajectories while maintaining sparse roadmaps are developed (Littlefield, Li, and Bekris 2013; Marble and Bekris 2013; Dobson and Bekris 2014; Salzman et al. 2014; Wang, Balkcom, and Chakrabarti 2015).

Shortest path query on graphs

Finding shortest paths on graphs has been studied extensively and Sommer gives a comprehensive survey (Sommer 2014).

4.2 Path convexity and subconvexity

In this work, we assume that an optimal steering method is available, together with a shortest-path local metric, d, that would describe the minimum cost of traveling between two configurations if there were no obstacles. The steering method and metric might be known exactly, or in practice, might be computed approximately using numerical techniques. Metrics are typically only available for symmetric systems.

The central idea is to cover most of the obstacle-free portion of the configuration space, C_{f}, with some convex closed cells, so that optimal paths cross through boundary points of the cells, and are well-behaved within the cells. However, for systems with a given metric, or subject to non-holonomic constraints, the straight lines used to define convexity in the usual sense might not be geodesics or even feasible.

It is natural to define convexity in terms of the steering method for the system. Let a steering method S of a metric space be a family of continuous curves (parametrized by arc length) such that for each ordered pair of points (a, b) in the metric space there is a corresponding curve in S from a to b.

We say that a set X of a metric space with steering method S is **path convex** if between any two points in X, the corresponding path from S is contained entirely within X. This definition is somewhat related to definitions of *geodesic convexity* proposed by Whitehead (Whitehead 1932).

As a simple example, let the metric space be the surface of a solid globe that has been cut in half through the equator, with the metric defined by the shortest path along the surface, measured using the Euclidean distance in R^3. Consider an optimal steering method S that contains one shortest path on the surface for each pair of points.

First consider the set F, the flat disc where the globe was cut. The shortest paths between points in F are straight lines, and they do not leave the disc, so F is path convex (under S). Now consider P, consisting of all points on the surface within 15

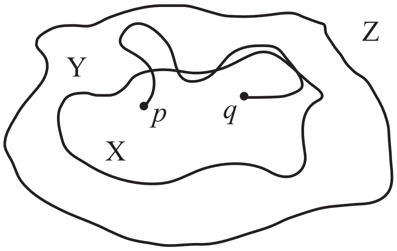


Figure 4.3: Subconvexity of set X with respect to set Y, under an optimal steering method. The path between p and q does not leave Y.

degrees of the pole. The shortest paths between points in P are arcs of great circles on the sphere and those arcs do not leave P, so P is path convex.

Finally, consider the set H, the entire curved hemisphere within 90 degrees of the pole. For some pairs of points in H, the connecting shortest path is an arc of a circle contained in H. But for other points (along the equator, for example), the connecting shortest path takes a shortcut through F, leaving H. So H is not a path convex set.

For the purpose of motion planning, we would like to be able to place a convex cell almost anywhere in the space, but for many systems, path convexity is too strong a requirement. We cannot cover H with path convex cells contained within H under an optimal steering method, so if there are obstacles in F, motion planning requires knowledge of F even if we care only about planning paths between points in H.

Given sets X and Y of a metric space, under a steering method S, we say that X is **path subconvex to Y under S** if between any two points in X, the corresponding path in S is contained within Y. For the purpose of motion planning, Y may be an obstacle-free region large enough to allow optimal maneuvering of the system between points of X.

A closed metric ball of radius c at a point x in a metric space Z under metric d, defined in the usual way as $B_c^d[x] = \{z \in Z : d(x, z) \leq c\}$, represents the set of points reachable from x with cost no greater than c. Although we might like to sample some points using closed metric balls at those points to cover some part of the free configuration space, we can see from the half-globe example (H is a closed metric ball centered at the pole) that a closed metric ball is not necessarily path convex with respect to an optimal steering method; the shortest path between points on the boundary might leave the ball, into regions we don't necessarily know anything about.

Although a closed metric ball (reachable set) at a sampled point might not be path convex, we can find a pair of balls at any point in C_f such that the inner, smaller ball is subconvex to the outer:

Theorem 4.1. *Given an optimal steering method S, a corresponding metric d over a metric space Z, a point $x \in Z$, and a positive constant r, the closed metric ball $B^d_{r/2}[x]$ is path subconvex to $B^d_r[x]$ under S.*

Proof. Consider two arbitrary points p and q in $B^d_{r/2}[x]$, and a postulated shortest path between them. Let m be an arbitrary point along this shortest path. We will show that $d(x, m)$ is no larger than r, implying that the entire path is contained within the larger ball. By the triangle inequality,

$$\begin{aligned} d(x,m) &\leq d(x,p) + d(p,m) && (4.1) \\ d(x,m) &\leq d(x,q) + d(q,m). && (4.2) \end{aligned}$$

Summing 4.1 and 4.2,

$$2d(x,m) \leq d(x,p) + d(x,q) + d(p,m) + d(m,q). \tag{4.3}$$

Since $d(x,p)$ and $d(x,q)$ are each less than or equal to $r/2$,

$$2d(x,m) \leq r + d(p,m) + d(m,q). \tag{4.4}$$

Since the path from p to q through m is a shortest path, $d(p,m) + d(m,q)$ must be less than or equal to the length of the path from p to q through x. Therefore, $d(p,m) + d(m,q) \leq r$. Combining with Inequality 4.4, $d(x,m) \leq r$. □

4.3 Computing reachable balls

From a particular configuration, how far, under a given metric, can the robot or system travel before hitting an obstacle? We expect this question to be hard to answer in the configuration space, since we do not typically know the shapes of configuration-space obstacles, and since we do not necessarily expect to even know the shapes of metric balls in configuration space. However, it is perhaps good enough to find a conservative estimate of the size of the safely-reachable ball, and this section will show that computing such an estimate might not be too difficult.

Many robotic systems of interest are embedded in two- or three-dimensional Euclidean workspaces, where geometric quantities like distance are much easier to measure. Recall a few classical definitions related to configuration space (Mason 2001). A *system* is a collection of particles embedded in a space, perhaps R^2 or R^3. We assume that the system is divided into two closed sets: the robot(s), which we control, and obstacles, which we don't. A *configuration* gives the locations of all particles. There are typically constraints on the possible configurations of the particles; the *configuration space* is the space of configurations satisfying the constraints.

In general, assume we have a configuration space parametrized by a vector $q \in R^n$ (or by an overlapping set of such parametrizations, an *atlas*), where n is the dimension of the configuration space. We will further assume that the parameters are bounded; let C_{f} be a bounded subset of R^n, or a finite collection of such subsets representing possible values of the parameters for which there is no collision.

In order to compute a lower bound on how much the configuration can change before a collision, we relax constraints on how particles can move. At a particular configuration, there is some minimum Euclidean distance from the obstacles, over all particles, $e(q)$. We also expect that, over all particles, and over all possible configurations, there is some maximum rate of change of location of any particle (measured by Euclidean distance), $v_{\max}$, with respect to a unit rate of change of the metric. Then define

$$d_{\mathrm{safe}}(q) = e(q)/v_{\max}, \tag{4.5}$$

the lower bound on the change in the metric before a collision; different sizes of cells are show in Figure 4.4.

4.3.1 Example: 2R planar arm

As a concrete example, consider a planar, serial robot arm with a fixed base, and two links that are each line segments of length one, shown in Figure 4.2. Let the configuration space be parametrized by $q = (\theta_1, \theta_2)$, and for simplicity, place joint limits such that each parameter falls in the range $[0, \pi]$.

We need a locally accurate metric d, which will describe distances in the configuration space with obstacles removed, together with an optimal steering method. Motivated by the observation that if each joint has the same constant upper bound

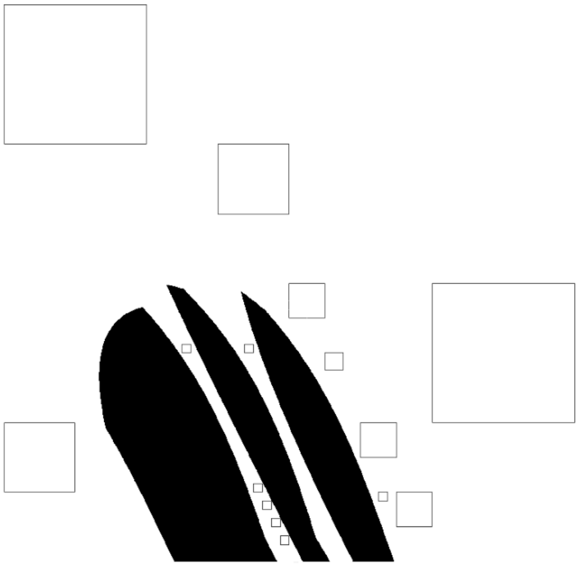


Figure 4.4: Different size cells at various configurations in the configuration space of the 2R arm.

on velocity, the time cost of moving from one configuration to another is determined by the joint that needs to move the farthest, define

$$d(q, q') = \max(|\theta_1' - \theta_1|, |\theta_2' - \theta_2|). \tag{4.6}$$

For this simple example, we know the shape of a metric ball: a square in the parameter space (θ_1, θ_2), possibly cut by the joint limits. One simple optimal steering method would be to move the joint that has farther to travel at maximum velocity towards the goal angle, and the other joint at a scaled rate, so that both joints reach their final angles at the same time: a line segment in the parameter space. The metric ball is path convex with respect to this steering method.

If we sample a point in the parameter space (θ_1, θ_2), how big of a ball can we place? We can compute the minimum Euclidean distance of points in the arm from the obstacles, $e(q)$, easily. We can also compute $v_{\max}$, the maximum rate of change of $e(q)$ over all trajectories in the configuration space, if configuration-space trajectories are parametrized by arc length measured by the metric. Notice that the “fastest" particle, over all particles, over all configurations, and over all unit-metric velocities of the joints is the point at the far tip of the arm, when the arm is fully extended, as the two joints move in the same direction at equal velocity. Simple differential kinematics indicate that the maximum speed of this particle is 3. So for this arm, $d_{\text{safe}}(q) = e(q)/3$.

4.3.2 Reachable balls for systems with Lipschitz continuity

In general, in order to ensure that a $v_{\max}$ exists that gives an upper bound on the rate of change of distances in the workspace with respect to the metric, we can verify that two properties hold for the system, metric, and steering method. First, that there exists a Lipschitz constant that bounds the workspace velocity (over all particles) with respect to change in configuration-space coordinates. Second, there exists a Lipschitz constant that bounds the rate of change of each of the configuration-space coordinates with respect to the metric along any path returned by the steering method. The upper bound $v_{\max}$ can then be computed as the product of the Lipschitz constants.

4.4 Safe covers of free configuration space

There are two structures of interest: the free configuration space C_{f}, which we are trying to approximate, and which will contain the union of “outer balls" described in Theorem 4.1, and a slightly smaller subset C_{ϵ}, all configurations that have distance at least $\epsilon > 0$ to obstacles in workspace, which we will actually cover with cells that are subconvex with respect to C_{f}. We will call this collection of cells a *safe cover*, since paths from the steering method between boundary points of the cells remain safely within C_{f}.

The shape of a metric ball in the configuration space is typically hard to obtain (although this was easy for the toy example of the serial arm), and it can be hard to understand how metric balls overlap and connect. We would also like to sample the configuration space efficiently, and not place new samples inside cells that have already been explored. Thus, in this section, we show how to construct a cover of C_ϵ by a set of hypercubes that only overlap along their boundaries; each hypercube is path subconvex with respect to a subset of C_{f}.

4.4.1 Reachable hypercubes

Given the maximum velocity, $v_{\max}$, of the robot over all configurations, at any configuration q not in contact with a workspace obstacle ($e(q) > 0$), we can compute a safe metric ball centered at q by Equation 4.5, and by Theorem 4.1, a subconvex inner ball with half the radius. We would like to place a hypercube entirely within this subconvex ball.

In general, given a ball of radius r centered at a configuration q, how large a hypercube (also centered at q) can we place within the ball? Let $c_h = \{p \in C_{\mathrm{f}} : p_i \in [q_i - h, q_i + h], \forall 1 \leq i \leq n\}$ be a hypercube with side length $2h$. We need to choose h so that every point of c_h is reachable in time (or more generally, metric cost) r.

For some systems, there may be particularly good methods for computing such an h. Here is a method which is perhaps more conservative than we might like, but which is a fairly general approach.

Let a coordinate-steering function S_i be a steering method that provides a path between a configuration q and a configuration $q + (0, 0, \ldots, x, 0, 0)$, such that the result of the motion is a change only in the ith parameter. (In the simplest case, each S_i might simply be the optimal steering method that we assume is available for the system, S.) Let $d_i(x)$ be the cost of applying steering method S_i in coordinate direction i to move a coordinate-distance x. For a symmetric system, if h satisfies

$$\sum_{i=1}^{n} d_i(h) \leq r, \tag{4.7}$$

and each $d_i(x)$ is non-decreasing in x, then $c_h \subset B_r$. We can find a suitable h value by binary search on h, recomputing costs of coordinate steering methods and adding them, and checking if 4.7 is satisfied.

For many systems, including systems without non-holonomic constraints, the Reeds-Shepp car, and kinematic differential-drive models, each $d_i(x)$ is non-decreasing. We must note that for non-STLC systems, such as the Dubins car, this is not the case.

4.4.2 Uniform grid cover

A simple approach to covering the space is to divide the space into cells of equal size placed on a uniform grid. We want the cells to cover C_ϵ, while being completely contained within (and at least subconvex to) C_f. The following observation indicates that small enough cubes, at configurations for which the workspace distance from obstacles is small enough, are not part of C_ϵ, and may be safely ignored.

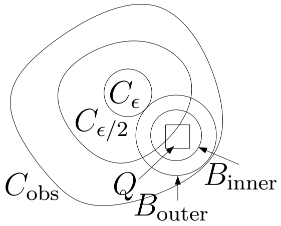


Figure 4.5: Illustration of proof of Theorem 4.2.

Theorem 4.2. *Let $e(q)$ be the minimum Euclidean distance to obstacles and $2h(e(q))$ be the side length of a subconvex hypercube inside the metric ball $B^d_{e(q)/v_{\max}}[q]$ computed in Section 4.4.1. For any $\epsilon > 0$, and any hypercube Q centered at q with side length $2s$, if $e(q) < \epsilon/2$ and $s < h(\epsilon/2)$, then $Q \cap C_\epsilon = \emptyset$.*

Proof. Since $e(q) < \epsilon/2$, the ball $B_{\text{outer}} = B^d_{\epsilon/(2v_{\max})}[q]$ is in C_{f}. Consequently, the ball $B_{\text{inner}} = B^d_{\epsilon/(4v_{\max})}[q]$ does not intersect with C_ϵ and is subconvex with respect to B_{outer}. If Q's side length is at most $h(\epsilon/2)$, then Q must lie in the ball B_{inner} by construction and hence $Q \cap C_\epsilon = \emptyset$; see Figure 4.5. □

Theorem 4.2 suggests a naïve algorithm: partition the configuration space by a set of hypercubes with side length $h(\epsilon/2)$ and discard all hypercubes with centers with Euclidean distance less than $\epsilon/2$ to obstacles in the workspace. Although this naïve algorithm creates a safe cover of C_ϵ, the algorithm is not efficient, since the algorithm generates hypercubes with a uniform size.

4.4.3 Cube subdivision cover

In order to obtain some larger hypercubes, we use a recursive approach: for a hypercube Q, if Q is subconvex with respect to C_{f}, then we keep this hypercube. Otherwise, subdivide Q into 2^n sub-hypercubes and find a cover recursively. By Theorem 4.2, we

```
Algorithm 2: cubeCover
  input  : Configuration space $C$, error parameter $\epsilon$
  output: A cover of $C_\epsilon$ by a set of hypercubes.
  Let $q$ be the center of $C$ and $2s$ be the side length of $C$.
  if $C$ passed subconvexity test then
      return a hypercube of $C$.
  else if $e(q) < \epsilon/2$ and $s < h(\epsilon/2)$ then
      return $\emptyset$.
  else
      Divide $C$ into $2^n$ hypercubes and recurse.
      return the union of all results.
```

can stop subdivision when the center has Euclidean distance less than $\epsilon/2$ and has a size smaller than $h(\epsilon/2)$. The result is a quadtree-like structure in n dimensions; Figure 4.2 shows a simple example for the 2R arm.

The crucial part is to test if some cell is subconvex with respect to C_{f}, without false positives. One way to test if a cell is subconvex is to compute a reachable ball centered at the center of the cell. Then divide the radius of this ball in half, to get a subconvex ball. Finally, compute the size h of a hypercube that fits in the smaller ball, and compare h to the size of the cell.

Algorithm 2 shows the approach, which builds the cell cover in depth-first order. Alternatively, one could explore the cover in a breadth-first order, so that large cells would be explored first, and the space could be constructed in an on-line fashion.

We use the 2R planar arm as an example. Since any hypercube in the free space of the arm is convex, testing the subconvexity of a hypercube only requires checking whether the hypercube is collision-free. Because reachable balls are themselves hypercubes for this simple example, if the side length of Q is smaller than $d_{\mathrm{safe}}(q)$, then Q must be collision-free. Thus, the decomposition algorithm can be applied very simply to the 2R planar arm system; see Figure 4.2.

4.4.4 Covering collision space

Even if sampling a hypercube center results in an actual collision, Algorithm 2 subdivides that cube, since the hypercube may contain smaller hypercubes that are collision free and convex, or subconvex to C_{f}, with respect to the steering method. This is problematic, since the effect is that all of collision space, the part of C inside obstacles, is divided into hypercubes of size that may be as small as $h(\epsilon/2)$, typically dominating computational and space costs for the algorithm.

Fortunately, there is a relatively easy solution. For each sampled hypercube center that results in a collision, we may compute a conservative bound on *penetration depth*: the minimum distance (as measured by d in the configuration space) required to escape the collision space. We use a technique similar to that used to compute bounds on distances to obstacles in configuration space. First, for a given configuration, we compute the maximum Euclidean distance, over all points on the robot, to the surface of the obstacle in the workspace, e_{escape}, a quantity analogous to $e(q)$. Then the penetration distance is at least equal to $d_{\mathrm{escape}} = e_{\mathrm{escape}}(q)/v_{\mathrm{max}}$.

If the current hypercube fits inside a ball of size d_{escape}, then the hypercube need not be subdivided. The hypercube may be discarded if we are interested only in the free space; in fact, we may add $\epsilon/2$ to $e_{\mathrm{escape}}(q)$, allowing many of the smallest hypercubes along the boundary be to discarded. Or, we may store these hypercubes if we are interested in problems that require information about topological properties of the obstacle space, such as proving non-existence of paths (McCarthy, Bretl, and Hutchinson 2012).

4.4.5 Larger cells, and cell merging

The computations of d_{safe} and of d_{escape} based on workspace information, as described above, are simple, fast, and too conservative. There are several ways in which the estimates may be improved, allowing larger cells.

First, $e(q)$ is the distance of the closest point on the robot to the obstacle, but the fastest point (with respect to the metric) may be some other point. So at some configuration, consider each point on the robot individually. For a given point p, compute the distance to the obstacles in workspace, $e(q, p)$. Now also compute the maximum speed for that point, $v_{\max}(p)$, over all configurations. For example, for the 2R arm, a point at the end of the first link would have a maximum workspace speed of one, not three. Divide to compute $d_{\text{safe}}(p)$, and choose d_{safe} as the minimum of $d_{\text{safe}}(p)$ over all points p on the robot.

Since we expect the robot to contain a continuum of points, finding this minimum can be difficult. We may, however, divide the robot up into pieces (for example, links on the arm), and for each piece, compute a shortest distance and highest speed. In our implementation of the hypercube cover algorithm for the arm, we took this approach, dividing the arm into links, computing d_{safe} for each link, and taking the minimum.

Second, computing $v_{\max}$ over all configurations is too conservative, since we are only interested in whether configurations near the current sample, q, may collide. For example, if the arm is folded back on itself ($\theta_2 = \pi$), then the maximum speed of any point should be much less than three in the local region. It is not clear how to directly compute a d_{safe} value taking this into account, since the maximum speed over the d_{safe} interval depends on the change in the configuration, and the change in the configuration depends on the speed and the length of the interval.

However, we can conduct an exponential search (Bentley and Yao 1976) for d_{safe}. First, assume that we're given some interval $\hat{d}$ and an initial configuration q. We can construct an oracle that will tell us whether $\hat{d}$ is either definitely safe, or possibly unsafe. For example, for the arm, we might choose motions that move the maximally extend the arm within the permitted interval, use the farthest point of extension to compute $v_{\max}$, and check if $\hat{d}v_{\max} > e(q)$. If so, the interval $\hat{d}$ is possibly unsafe; divide $\hat{d}$, and repeat until a satisfactory (but conservative) approximation of d_{safe} value has been found. We used such an exponential search in the example hypercube decompositions for the nR arms below.

Finally, it should be noted that adjacent free convex cells may be merged into a single cell, and we used this fact to merge some cells for planar arms. Interestingly, adjacent subconvex cells cannot necessarily be merged; we do not merge adjacent cells for the Reeds-Shepp car example, but would like to understand better how compression of groups of such adjacent cells might be done.

4.5 Existence of optimal paths in the cover

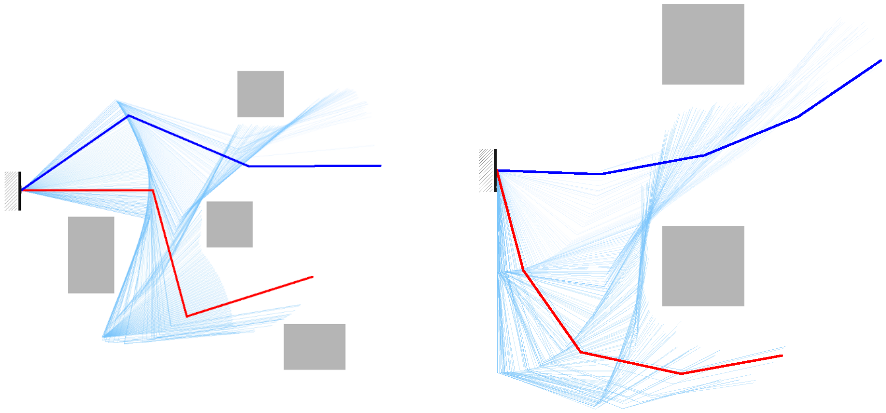
Sometimes, optimal trajectories do not exist in the free configuration space; for example, Desaulniers showed that for the Reeds-Shepp car among simple polygonal obstacles, there can exist pairs of configurations such that given any connecting trajectory with a finite number of control switches, there is a shorter connecting trajectory: chattering (Desaulniers 1996). However, optimal trajectories exist in finite sets of subconvex cells and chattering does not occur:

Theorem 4.3. *Given a pair of points contained within a finite, connected set of cells subconvex to the free configuration space, with corresponding metric that is continuous along the boundaries of each cell, there exists an optimal trajectory that can be described by a non-repeating sequence of cells $c_1, \ldots, c_m$, together with one point $p_i \in c_i$ for each cell except the last, such that this optimal trajectory consists of optimal trajectories connecting the start to p_1, p_1 to p_2, etc, and finally from p_{m-1} to the goal.*

Proof. Since the cells are connected, there is a set of curves that connect the given points; let these curves be parameterized by arc length under the given metric. Any such connecting curve will enter a cell for a first time, and exit each cell (except the last) for a last time, s_l. If we replace this section of the curve with an optimal trajectory within the cell, the new curve will certainly be no longer than the original.

Thus, to find an optimal curve, it is sufficient to consider the set of curves connecting the start and goal, with one locally optimal section per cell. Each such curve can be described by the discrete structure of the curve (the non-repeating sequence of cell indices, sorted by increasing order of s_l values), together with parameters describing the configurations at which the curve leaves each cell for the last time. Since there are finitely many structures (upper bounded by the cardinality of the power set over the set of cells), we need only show that for each structure, an optimal trajectory exists.

The problem of finding the minimum cost for a particular structure is a finite-dimensional optimization (there are finitely many parameters describing final exits from cells) over a compact set (the boundaries of the cells) of a continuous function (the finite sum of metric functions over the cells); an optimal solution therefore exists,



(a) 3R planar arm (b) 4R planar arm

Figure 4.6: Example trajectories of 3R and 4R planar arms.

as does a corresponding optimal trajectory. □

We expect Theorem 4.3 to be useful for designing motion planning algorithms: no cell ever needs to be visited twice, although both the sequence of cells and the final exit points from each need to be determined.

4.6 Case study: planar arms

To find a trajectory connecting an initial configuration q_0 and a goal configuration q_g, we create samples along the boundary of all hypercubes at some resolution. We use the steering method to compute the distance between every pair of samples in the same hypercube. Then, we find the shortest path in the graph induced by these samples, using a straightforward A* search.

The heuristic function we used is a lower bound on the distance between the sample represented by the state to the goal, which can be easily computed based

scenario	ϵ	number of cells	running time in seconds
Figure 4.6a	0.2	1350697	148
Figure 4.6b	0.3	4630896	332

Table 4.1: Performance for robot arm system.

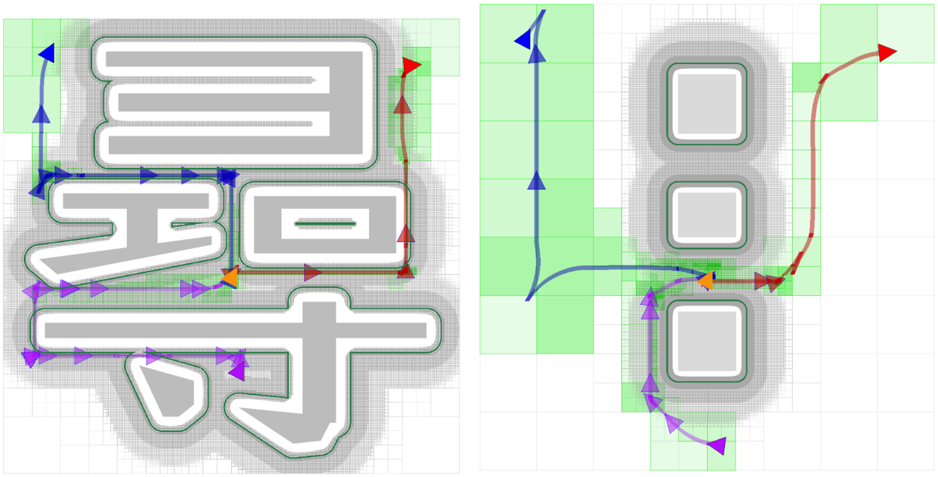
on the Euclidean distance in the workspace and the maximum velocity. For specific systems, we expect designing a more accurate heuristic function to be possible.

We take the 3R planar arm 4R planar arm as test cases for the planning algorithm; two example resulting trajectories are shown in Figure 4.6. We implement the algorithm in C++ and conducted tests on a modern desktop machine (iMac) with an Intel Core i5 2.7 GHz CPU and 16GB RAM. Table 4.1 shows the running times, memory, and number of cells generated while constructing representations of c-space for the figures shown. Most of time is spent in the collision detection, since we only use an elementary method to check collision.

4.7 Case study: Reeds-Shepp car

In this section, we show how to apply the decomposition algorithm to other systems, in which metric balls are hard to compute, including non-holonomic systems, using the Reeds-Shepp car as an example. Remember that the crucial part of the decomposition algorithm is to test a given hypercube Q is subconvex with respect to C_{f} without false positive. We gave a general approach based on Lipschitz continuity of the steering method in Section 4.4. However, the resulting hypercubes tend to be smaller than we would like.

Here, we give another (numerical) procedure for testing the subconvexity of a hypercube. Compute the swept volume in the workspace for all trajectories connecting



(a) (b)

Figure 4.7: Several example trajectories in different environments of the Reeds-Shepp car. Orange triangles indicate common goal configurations. Shaded polygons are obstacles in workspace. Solid curves around obstacles are the boundary C_ϵ in work space. Subconvex hypercubes are projected into workspace as cubes. Dark green boundaries show the grown obstacles in the workspace.

all pairs of configurations in the hypercube. If the swept volume is collision free, then this hypercube is subconvex by definition.

Computing this swept volume analytically is difficult for most systems. Thus, instead of computing swept volume analytically, we approximated the swept volume numerically in the following way: first, densely sample configurations within the hypercube and use the steering method to compute trajectories between all pairs of configurations. Second, compute an approximate bounding volume for all trajectories in the workspace. If the bounding volume is collision free, then this hypercube can be considered subconvex. By using this numerical testing procedure, the size of the subconvex hypercube found by the decomposition algorithm can be greatly increased.

Optimal trajectories for the Reeds-Shepp car can be found analytically, and we

scenario	ϵ	number of cells	running time in seconds
Figure 4.7a	0.2	24216563	312
Figure 4.7b	0.15	25218358	66

Table 4.2: Performance for Reeds-Shepp car.

use the optimal trajectory solver as the steering method (Laumond 1998). The configuration space to be $[-\pi, \pi]^3$ and the car is represented as an isosceles triangle with base length 0.25 and height 0.25. The resulting trajectories are shown in Figure 4.7 and the performance of Algorithm 2 for 4R planar arm system is in Table 4.2.

4.8 Conclusions and future work

We have presented a definition of convexity that we believe is useful for understanding the interplay between local, optimal steering methods and the global structure of the configuration space. This is our first work on this problem, and we have not yet conducted exhaustive experimental exploration of the properties of the cell decompositions described. Initial results are promising, however, in that for low-DOF systems, we are able to construct apparently very good trajectories from constructed cell-decompositions, and cells corresponding to large open spaces in the workspace are quite large.

It may not be too surprising that a very large number of cells is needed to represent the area of the configuration space near obstacles, or that we do not escape the curse of dimensionality. We intend to explore methods, perhaps exploiting properties such as visibility, that allow sparser representations near obstacles while still allowing some approximation guarantees about optimality to be maintained. We would also like to explore topological properties of configuration spaces using these cell decompositions, along the lines of recent work by Bhattacharya *et al.* (Bhattacharya et al. 2013).

Chapter 5

Maximum k-survivability

5.1 Introduction

How should a set of robots move through a dangerous environment to accomplish objectives? Is it better for the robots to travel together, or should the robots split up? What is the relationship between survival and diversity of actions?

As an example, consider the following whimsical planning problem: n ants must migrate from one nest to another through a field containing both obstacles and antlions, which make disc-shaped traps. If we assume a uniform distribution of trap locations, which n paths should the ants follow, if the ants must decide their paths before moving and cannot reroute during movement?

One idea might be to maximize the expected number of surviving ants. However, the best strategy for this problem turns out to be uninteresting and unwise: find the safest path for a single ant (for simplicity, assume there is a unique safest path), and have all ants follow that path. This solution is not robust – a single trap could destroy the entire colony. Therefore, we consider a problem that is more suitable if ants are

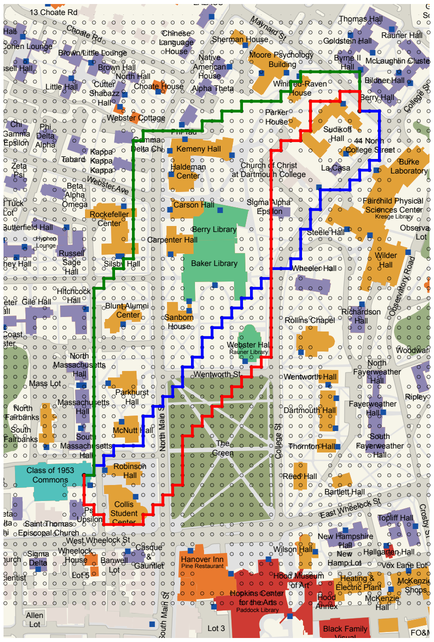


Figure 5.1: Toy example of a high 1-survivability set of three paths from Department of Computer Science (Sudikoff) to food court (1953 Commons) on Dartmouth College campus, where gray circles represent the discretization of the environment.[1]

expendable: maximize the probability that at least some k (with $k \leq n$) ants survive. If the number of traps is unknown, the solution may contain up to n unique paths.

Figure 5.1 shows an example problem for which paths have been selected to achieve high survivability of routes across a college campus. The paths are short, interestingly diverse, and may be of practical interest if there is actual danger, traffic congestion, or surveillance to be avoided.

We believe this to be the first work that explicitly studies the theoretical implications of robot expendability. Path diversity has been explored in several settings, with applications including motion planning (Branicky, Knepper, and Kuffner 2008; Knepper and Mason 2009; Green and Kelly 2011; Knepper 2011; Knepper, Srinivasa, and Mason 2012; Quispe, Kunz, and Stilman 2013; Voss, Moll, and Kavraki 2015), robust routing in computer networks (Xu et al. 2006), and dissimilar paths in transportation (Dell'Olmo, Gentili, and Scozzari 2005). Approaches to finding diversity typically involve defining an arbitrary distance metric that describes separation of paths, and finding solutions that balance distance between paths against length of paths, using linear or non-linear weights, constrained optimization, or by analyzing the Pareto frontier.

Instead of defining an arbitrary pairwise path diversity metric or choosing arbitrary tradeoffs between path lengths and diversity metrics, our approach proceeds directly from the threat model, since we believe that diversity should be considered as a means rather than an end.

We define *k-survivability* to measure the quality of paths in a stochastic threat environment. Sets of paths with high k-survivability naturally balance length and diversity. Although choosing sets of paths to maximize k-survivability is NP-hard, we design a complete algorithm. Since the maximization algorithm is computationally infeasible except for $k = 1$ and $n = 2$, we also design a practically faster heuristic method that finds paths with high k-survivability.

5.1.1 Model

We focus on the discrete problem in which the environment is represented by a point set D and the free space is represented by a graph $G = (V, E)$, where $V \subseteq D$. Unknown stochastic dangers are called *traps*. Several models of traps will be discussed in the next section.

[1] The campus map is from Dartmouth College website and the street data is from OpenStreetMap. Since the data from OpenStreetMap is inconsistent with the campus map, some grids are inside buildings on the map.

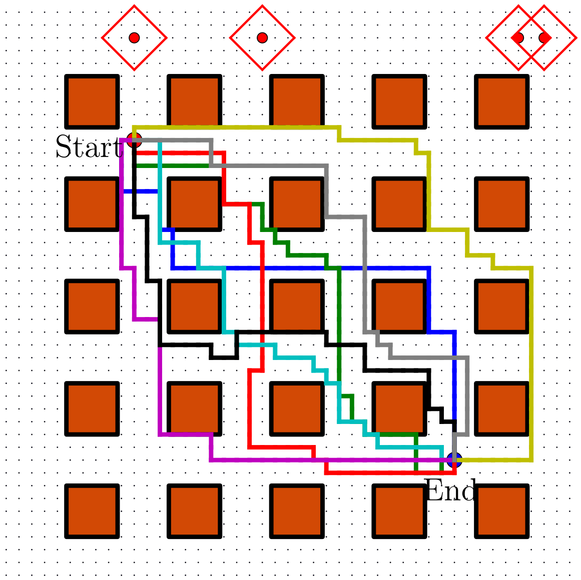


Figure 5.2: Eight paths with high 1-survivability. Small gray circles are vertices of G (4-connected). Squares are obstacles; diamonds are example fixed traps.

Our problem is to find paths for n point robots such that the i-th path connects the designated start vertex $s_i \in V$ and the designated goal vertex $g_i \in V$; see Figure 5.2. Robots cannot communicate, do not have sensors, and cannot reroute; both obstacles and traps are time-independent.

We define *k-survivability* to be the probability that at least k paths successfully connect their (perhaps different) starts to goals. The *k-survivability problem* (k**SP**) is formalized as:

Input $= (G, M, \{(s_i, g_i)\}_{i=1}^n, k)$, where

(a) $G = (V, E)$ denotes the free space.

(b) M is a trap model (see next section).

(c) n point robots have start locations $s_i \in V$ and goal locations $g_i \in V$ for all $1 \leq i \leq n$.

(d) survivability parameter k, with $1 \leq k \leq n$.

Output $= \mathcal{P}$, a list of n paths maximizing k-survivability such that for all $1 \leq i \leq n$, $P_i \in \mathcal{P}$ connects s_i and g_i.

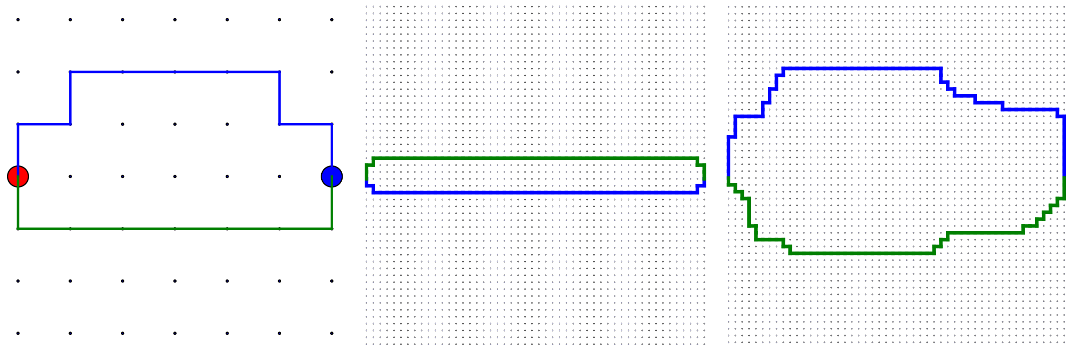
Trap models

We now discuss two trap models: *fixed traps*, which have known shapes, and *variable traps*, for which the shape is unknown but drawn from some known distribution. Even under the fixed trap model, a different trap shape (or even multiple traps) may be placed at each vertex.

Fixed traps. A *fixed trap* F is a subset of D. When a fixed trap F is *in effect*, all paths passing through F are blocked. A *fixed trap model* $M = \{(F_i, p_i)\}_{i=1}^{|M|}$ is a collection of fixed traps and their corresponding, independent probabilities. If all probabilities are equal, then the model is a *uniform fixed trap model.*

For example, under a *uniform fixed r-disc trap model*, each vertex has equal and independent probability to be the center of a disc trap of radius r. Figures 5.3a and 5.3b show examples.

Variable traps. A *variable trap* A is represented by a distribution over a set of fixed traps. A *variable trap model* M is represented by a collection of variable traps and corresponding probabilities: $M = \{(A_i, p_i)\}_{i=1}^{|M|}$. For example, under a *variable r-disc trap model*, each vertex has identical and independent probability to be the center of a disc trap, whose radius follows a geometric distribution with mean r. Two



(a) Paths with the highest 1-survivability under the fixed 1-disc trap model.

(b) Paths with high 1-survivability under the fixed 3-disc trap model.

(c) Paths with high 1-survivability under the variable 5-disc trap model.

Figure 5.3: Example paths for two robots in different environments and parameters. In Figure 5.3a, since $r = 1$, the optimal solution has parallel subpaths with distance two to avoid being destroyed by one 1-disc easily.

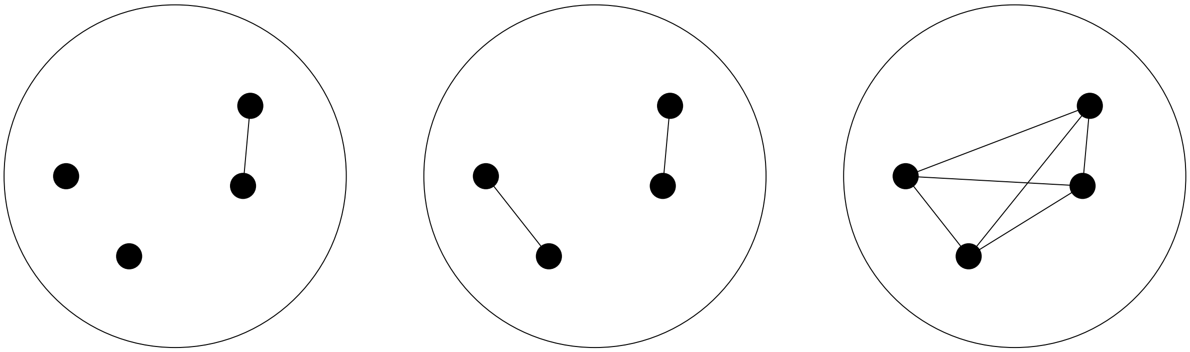
paths with high 1-survivability under the variable 5-disc trap model are shown in Figure 5.3c.

5.2 Related work

We briefly summarize related work in location theory, motion planning, graph theory, computer networks, and transportation.

The maximum diversity problem in location theory

In location theory, the maximum diversity problem is to find m points maximizing diversity among given points in a metric space. Although location theory researchers are more interested in finding diverse points, their methods can be adapted to find diverse paths as long as a metric space on paths can be defined. Formulations include



(a) Remote-edge: distance of the closest-pair.

(b) Remote pseudoforest: sum of the distances to nearest neighbors.

(c) Remote-clique: sum of mutual distances.

Figure 5.4: Illustration of maximum diversity problems. Each point in the circle represents a trajectory. Maximum diversity problem is to choose k trajectories so that the objective function is maximized.

(Chandra and Halldórsson 2001):

(a) *remote-edge problem*: find a set of points maximizing the minimum mutual distance (also called *p-dispersion* problem, Kuby 1987). There is a 2-approximation algorithm for remote-edge problem (Tamir 1991; White 1991) and finding a better approximation algorithm will imply $P = NP$ (Ravi, Rosenkrantz, and Tayi 1994).

(b) *remote-pseudoforest problem*: find a set of points maximizing the sum of the distance to the nearest neighbors (also called *p-defense* problem, Moon and Chaudhry 1984). There is an $O(\lg n)$-approximation algorithm for remote-pseudoforest problem (Chandra and Halldórsson 2001) and finding an approximation algorithm with approximation ratio smaller than 2 will imply $P = NP$ (Halldórsson et al. 1999).

(c) *remote-clique problem*: find a set of points maximizing the sum of mutual distances (also called *max-avg facility dispersion* problem, Ravi, Rosenkrantz, and Tayi 1994; or *maximum dispersion* problem, Kuby 1987). There is a 2-approximation algorithm for remote-clique problem (Hassin, Rubinstein, and Tamir 1997).

Since the maximum diversity problem is difficult, several heuristics have been proposed; Martí *et al.* give a comprehensive survey for heuristics and metaheuristics for the maximum diversity problem (Martí et al. 2013).

Diverse trajectories generation in motion planning

Increasing the diversity of trajectories in motion planning has been studied by several researchers (Branicky, Knepper, and Kuffner 2008; Knepper and Mason 2009; Green and Kelly 2011; Knepper 2011; Knepper, Srinivasa, and Mason 2012; Quispe, Kunz, and Stilman 2013; Voss, Moll, and Kavraki 2015). Green's and Kelly's approach is to consider a distance semi-metric space (X, δ) of trajectories and they define the diversity of trajectories as the minimum distance between trajectories (Green and Kelly 2011). On the other hand, Knepper and Mason define *path space* to be a metric space of trajectories (Knepper, Srinivasa, and Mason 2012).

Our work is most related to Erickson's and LaValle's work (Erickson and LaValle 2009). They propose a definition of *survivability* that measures the correlation of damage on paths when a random disc obstacle is placed on a path. Whereas survivability favors separated paths, k-survivability is a direct probabilistic measure of survival that in some cases can be maximized by allowing robots to follow overlapping short paths.

Trajectory diversity in the Euclidean plane. In the Euclidean plane, finding a trajectory connecting two points among polygonal obstacles can be solved efficiently (Hershberger and Suri 1999). One possible definition for the diversity of trajectories in the Euclidean plane is the number of distinct homotopy classes of trajectories (Bhattacharya, Likhachev, and Kumar 2012). Finding shortest paths subject to topological constraints has been studied (Hershberger and Snoeyink 1994; Bespamyatnikh 2003; Efrat, Kobourov, and Lubiw 2006; Verdière and Erickson 2010). Eriksson-Bique *et al.* studied the problem of finding k shortest trajectories with distinct homotopy classes (Eriksson-Bique et al. 2015).

Single trajectory in a threat environment. Finding trajectories in a threat environment has been studied for aircrafts (Zabarankin, Uryasev, and Murphey 2006), UAVs (Miller et al. 2011), vehicles (Boidot, Marzuoli, and Feron 2015), and ships (Babel and Zimmermann 2015). Our work differs in that the threat model is probabilistic, and in the search for multiple trajectories.

Path diversity on graphs

Finding a shortest path from one vertex to another on a graph is a fundamental problem in combinatorial optimization. One possible definition for the diversity of paths on graphs is the number of different paths. The problem of finding k-shortest paths on a graph has also been well-studied (Yen 1971; Eppstein 1998; Hershberger, Maxel, and Suri 2007; Gotthilf and Lewenstein 2009; Roditty 2010; Aljazzar and Leue 2011; Frieder and Roditty 2014).

Since paths sharing an edge may be vulnerable, researchers tried to find (vertex/edge) disjoint paths. The problem of finding shortest disjoint paths has been considered (Suurballe 1974; Fortune, Hopcroft, and Wyllie 1980; Suurballe and Tarjan 1984; Li, McCormick, and Simchi-Levi 1990; Li, McCormick, and Simchi-Levi 1992; Eilam-Tzoreff 1998; Bhatia, Kodialam, and Lakshman 2006). Disjointness seems a strong constraint since path length may increase significantly, so several relaxations of disjoint paths problem have been studied. For example, the problem of finding k paths minimizing shared edge (Omran, Sack, and Zarrabi-Zadeh 2013), the problem of finding k shortest paths without sharing more than r edges (Assadi et al. 2014), and the problem of finding the minimum set of paths to ensure shortest path is available under any k edges failure (Zhang, Xu, and Wen 2015) have been studied recently.

When each vertex/edge has a failure probability, short and reliable paths are desirable. Finding a shortest path subject to reliability constraints can be considered as resource-constrained shortest-path problems (Joksch 1966). One problem that is related to maximum arrivability problem is the *reliable two-path problem with arc sharing* (RTP-S, Andreas and Smith 2008). The difference between maximum k-survivability problem and RTP-S is that the maximum k-survivability problem treats both length and survivability in the objective function but RPT-S treat the survivability, which they called reliability, as a constraint.

Path diversity in network

One goal of network design is robustness and one way to improve the robustness of a network is to increase the path diversity between end-points (Bhandari 1999). Diverse routing problems has been studied for more than a decade using graph theory (Cidon, Rom, and Shavitt 1999; Brumbaugh-Smith and Shier 2002; Ho, Tapolcai, and Cinkler 2004; Xu et al. 2006; Stefanakos 2008; Zotkiewicz, Ben-Ameur, and Pióro 2010; Zheng et al. 2010; Lee, Modiano, and Lee 2010; Yuan and Wang 2011; Gomes and Zotkiewicz 2014). Rohrer *et al.* define the diversity of paths by using the distance on graphs and geographic distances (Rohrer, Jabbar, and Sterbenz 2014), which is similar to the idea of path space (Knepper, Srinivasa, and Mason 2012).

Dissimilar paths in transportation

The problem of finding dissimilar paths has been studied in transportation, since dissimilar paths are beneficial for hazardous waste transportation for safety reasons (Akgün, Erkut, and Batta 2000; Dell'Olmo, Gentili, and Scozzari 2005).

5.3 Computing k-survivability

Since k-survivability is independent of the order of vertices along paths, paths are represented as sets of vertices.

5.3.1 Computing k-survivability under the fixed trap model

Given a fixed trap model $M = \{(F_i, p_i)\}_{i=1}^{|M|}$ and a path P on a graph G, the *forbidden index set* of P is $\mathrm{Forbid}(P) = \{i \mid P \cap F_i \neq \emptyset\}$. The probability that P is not blocked equals $\Pr(P) = \prod_{i \in \mathrm{Forbid}(P)} (1 - p_i)$. Similarly, for a set of paths $\mathcal{P} = \{P_1, \ldots, P_h\}$, the *forbidden index set* of $\mathcal{P}$ is $\mathrm{Forbid}(\mathcal{P}) = \cup_{P \in \mathcal{P}} \mathrm{Forbid}(P)$. The probability that all paths in $\mathcal{P}$ are not blocked equals $\Pr(\mathcal{P}) = \prod_{i \in \mathrm{Forbid}(\mathcal{P})} (1 - p_i)$.

Given a set of paths $\mathcal{P}$, 1-survivability can be computed by using the inclusion-exclusion principle as follows:

$$\text{Survive}_1(\mathcal{P}) = \sum_{\mathcal{S}\subseteq\mathcal{P},\mathcal{S}\neq\emptyset} (-1)^{|\mathcal{S}|+1}\Pr(\mathcal{S})$$

The computation of k-survivability is similar but more time-consuming. For a set of paths $\mathcal{P}$ and a positive integer k, we define $\text{comb}(\mathcal{P}, k) = \{\mathcal{S} \mid \mathcal{S} \subseteq \mathcal{P}, |\mathcal{S}| = k\}$; that is, $\text{comb}(\mathcal{P}, k)$ contains all k-subsets of $\mathcal{P}$. Given a set of paths $\mathcal{P}$, k-survivability can be computed by using the inclusion-exclusion principle:

$$\text{Survive}_k(\mathcal{P}) = \sum_{\mathbf{R}\subseteq\text{comb}(\mathcal{P},k),\mathbf{R}\neq\emptyset} (-1)^{|\mathbf{R}|+1}\Pr(\cup_{\mathcal{S}\in\mathbf{R}}\mathcal{S})$$

5.3.2 Computing k-survivability under the variable trap model

Let $M = \{(A_i, p_i)\}_{i=1}^{|M|}$ be a variable trap model. Each variable trap A_i can be represented as a collection of fixed traps and their probabilities: $A_i = \{(F_{i,j}, p_{i,j})\}_{j=1}^{|A_i|}$. Given a path P on a graph G, the *forbidden index set* of P with respect to the i-th variable trap is $\text{Forbid}(P, i) = \{j \mid P \cap F_{i,j} \neq \emptyset\}$. The probability that P is not blocked equals $\Pr(P) = \prod_i (1 - \sum_{j\in\text{Forbid}(P,i)} p_{i,j})$. The calculation of k-survivability can be derived in the same manner as under the fixed trap model.

5.4 Theoretical results

In this section, we show that kSP is NP-hard. We also show that maximizing k-survivability leads to diversity in the sense that at most k robots will follow the same path, if other paths are available.

5.4.1 NP-hardness of kSP

We show that kSP under the uniform fixed trap model is NP-hard with $n = k = 1$. Since the fixed trap model is a special case of the variable trap model, kSP under the variable trap model is also NP-hard.

Our proof, similar to the NP-completeness proof of the minimum color path problem (Yuan, Varma, and Jue 2005), is a reduction from the *minimum set cover problem* (MSCP), defined below, to kSP.

> **Input** $= (S, \mathcal{C})$, where
>
> (a) $S = \{1, \ldots, a\}$ is a set of positive integers from 1 to a.
>
> (b) $\mathcal{C} = \{C_1, \ldots, C_b\}$ is a collection of subsets of S.
>
> **Output** $= \mathcal{C}' \subseteq \mathcal{C}$ a minimum cardinality collection of subsets whose union is S.

Theorem 5.1. *kSP under the uniform fixed trap model with $n = k = 1$ is NP-hard.*

Proof. Let $(S, \mathcal{C})$ be an instance of MSCP. We construct an instance $(G, M, \{(s, g)\}, 1)$ of kSP in polynomial time such that an optimal solution in kSP can be transformed into a minimum set cover of $(S, \mathcal{C})$ and vice versa.

We construct $G = (V, E)$, which is a subgraph of a grid graph, as follows. First, for each element $i \in S$, we create three vertices $v_i^{0,0} = (4i, 0)$, $v_i^{1,0} = (4i + 1, 0)$, and $v_i^{3,0} = (4i + 3, 0)$. We create edges $(v_i^{0,0}, v_i^{1,0})$ for all $1 \le i \le a$ and edges $(v_i^{3,0}, v_{i+1}^{0,0})$ for all $1 \le i < a$. Our idea is to design a gadget for each element $i \in S$ such that i can be covered by a set in $\mathcal{C}$ if and only if a solution of kSP, which is a path, passes through $v_i^{0,0}$ and $v_i^{3,0}$.

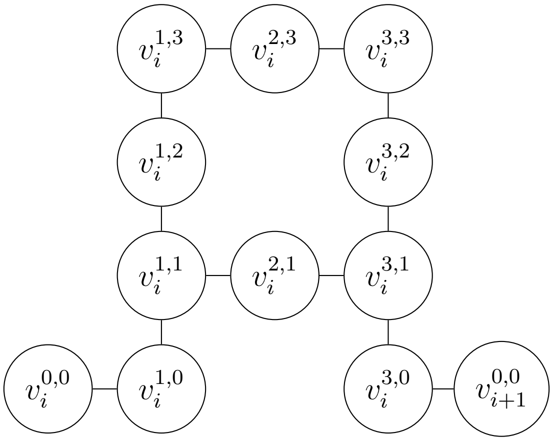


Figure 5.5: Gadget used in the proof of Theorem 5.1.

Second, for each $j \in \{1, \dots, b\}$, we create vertices $v_i^{1,j} = (4i + 1, j)$ and $v_i^{3,j} = (4i + 3, j)$ for all $1 \leq i \leq a$. Then, we create edges $(v_i^{1,j}, v_i^{1,j+1})$ and $(v_i^{3,j}, v_i^{3,j+1})$ for all $0 \leq j < b$. Intuitively, the j-th row represent the j-th set in $\mathcal{C}$.

Finally, we create vertices $v_i^{2,j} = (4i + 2, j)$, edges $(v_i^{1,j}, v_i^{2,j})$, and edges $(v_i^{2,j}, v_i^{3,j})$ for each $i \in C_j$. We use these vertices to model the constraint that every element in S is covered by a set in $\mathcal{C}$; see Figure 5.5.

The uniform fixed trap model is $M = \{(F_j, p)\}_{j=1}^{b}$ for an arbitrary choice $p \in (0, 1)$, where $F_j = \{v_i^{2,j} \mid i \in C_j\}$. The start vertex is $v_1^{0,0}$ and the goal vertex is $v_a^{3,0}$.

Let P be an optimal solution of the instance $(G, M, \{(v_1^{0,0}, v_a^{3,0})\}, 1)$ of kSP. By the construction of the graph, P passes every vertex $v_i^{3,0}$ for all $1 \leq i \leq a$. Moreover, for each $1 \leq i \leq a$, the only way to get $v_i^{3,0}$ is to pass through a vertex $v_i^{2,j}$ for some $1 \leq j \leq b$. Since $v_i^{2,j}$ exists if and only if $i \in C_j$, the set $\mathcal{C}' = \{C_j \mid \exists_{1 \leq i \leq a} P \text{ passes } v_i^{2,j}\}$ is a set cover of S.

Moreover, since all traps have the same probability, maximizing 1-survivability is the same as minimizing $\{j \mid \exists_{1 \leq i \leq a} P \text{ passes } v_i^{2,j}\}$. Hence, $\mathcal{C}'$ is also an optimal solution of MSCP.

Transforming an optimal solution of MSCP to an optimal solution of kSP can be done similarly. □

Note that this reduction relies on the fact that individual fixed traps might be formed from disconnected sets of vertices. However, even if we restrict individual fixed traps to be contiguous, the problem still appears to be hard, since kSP under the uniform fixed trap model with $n = k = 1$ can be used to solve the *barrier resilience problem* (Chan and Kirkpatrick 2014). The complexity of the barrier resilience problem is still open and currently no polynomial time algorithm exists.

Even approximating an optimal solution is hard:

Theorem 5.2. *No polynomial time algorithm with constant approximation ratio for kSP with $n = k = 1$ under the uniform fixed trap model exists unless $P = NP$.*

Proof. We show that if a r-approximation polynomial time algorithm for kSP exists for some constant r, then we can solve MSCP in polynomial time.

Let $(S, \mathcal{C})$ be an instance of MSCP and c be the size of a minimum set cover. By using the same reduction as in the proof of Theorem 5.1, we obtain an instance $(G, M, \{(s, g)\}, 1)$ of kSP. Since c is the size of a minimum set cover, the optimal solution of $(G, M, \{(s, g)\}, 1)$ has value $(1 - p)^c$.

Suppose that a r-approximation algorithm for kSP exists, $0 < r < 1$, and this approximation algorithm is guaranteed to find a solution with 1-survivability at least $r(1-p)^c$. Since the choice of probability p in the reduction is arbitrary, we set p to be a value satisfying $r > (1-p)$. Because $r(1-p)^c > (1-p)^{c+1}$ and a path can only pass through an integral number of fixed traps, the approximation algorithm must return a solution with value $(1 - p)^c$, which is an optimal solution of $(G, M, \{(s, g)\}, 1)$ and can be transformed into an optimal solution of $(S, \mathcal{C})$ in polynomial time. □

5.4.2 Properties of kSP

We now show that k-survivability leads to diverse paths.

Observation 5.3. For kSP under the fixed trap model with $k = 1$, if n paths with different forbidden index sets exist, then any optimal solution does not have duplicate paths.

Proof. Since paths with the same forbidden index sets are either all-safe or all-blocked, using paths with different forbidden index sets improves 1-survivability. □

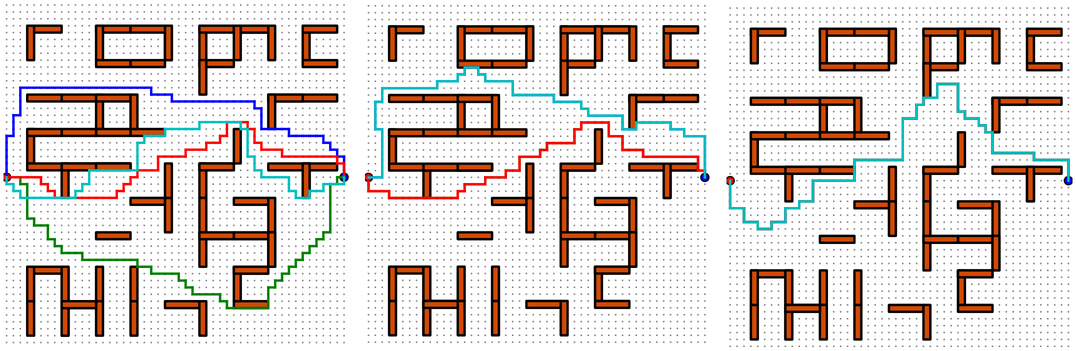
Observation 5.4. For kSP under the fixed trap model, if at least $\lceil n/k \rceil$ paths with different forbidden index sets exist, then at most k robots follow the same path in an optimal solution.

Proof. If more than k robots follow the same path, moving one robot to another path always improves k-survivability. □

Note that when k increases, the number of different paths in optimal solutions may decrease. See figure 5.6, which shows some high-survivability paths for different values of k.

5.5 Algorithms

In this section, we first describe a complete algorithm to find optimal solutions. Then, we describe a typically faster heuristic method to find good solutions. For simplicity, we assume that all robots have the same start s and the same goal g, but this assumption may be easily lifted.



(a) $k = 1$. All robots take different paths.

(b) $k = 2$. Robots form two groups.

(c) $k = 4$. All robots take the same path.

Figure 5.6: Example paths with high k-survivability for different values of k under the fixed 5-disc trap model.

5.5.1 Complete algorithm

In this section, we design a complete state space search algorithm for kSP under the uniform fixed trap model with $n = 2$ and $k = 1$. Although it is easy to extend this algorithm for larger k and n, solving even small problems becomes computationally infeasible with this approach.

We need several definitions. A path P is an ordered list of vertices. A path P' *extends* another path P, if P is a prefix of P'. A path P' is a *feasible extension* of P if either P ends at g and $P' = P$, or P does not end at g and P' extends P by one vertex. Let $\text{Ext}(P)$ denote the set of all paths that end at g and are extensions of path P.

The complete algorithm is a state space search algorithm. Each state t consists of two simple paths (P_1, P_2) starting from s. A state (P_1', P_2') is a *successor* of a state (P_1, P_2) if P_1' and P_2' are feasible extensions of P_1 and P_2 respectively.

The initial state is $(\langle s\rangle, \langle s\rangle)$ and the goal states are all states (P_1, P_2) that both P_1 and P_2 end at g. We will find one goal state with maximum 1-survivability.

Since the state space is a tree, we can use a brute-force approach to traverse the tree to find an optimal solution. In order to speed up the brute-force approach, we design a heuristic function h of states, where $h(t)$ is an upper-bound of 1-survivability of all goal states that are reachable by state t. As long as $h(t)$ is optimistic, then the tree search will find an optimal solution. Using the heuristic function, we can prune unnecessary branches and stop search when the algorithm reaches one of the goal states for the first time.

We construct a heuristic function h as follows. Remember that when $n = 2$, 1-survivability of two paths P_1 and P_2 is $\Pr(\{P_1\}) + \Pr(\{P_2\}) - \Pr(\{P_1, P_2\})$. Suppose that there is a function $\hat{h}$ for paths that $\hat{h}(P)$ is an upper bound of $\Pr(\{P'\})$ for all $P' \in \mathrm{Ext}(P)$. Then, we obtain a heuristic function $h((P_1, P_2)) = \hat{h}(P_1) + \hat{h}(P_2) - \Pr(\{P_1, P_2\})$.

Now, we show how to construct a function $\hat{h}$. Let M be the uniform fixed trap model. For any path P, 1-survivability of P is $(1-p)^{|\mathrm{Forbid}(P)|}$, which only depends on the size of $\mathrm{Forbid}(P)$. Let $\mathrm{LB}(P)$ be the minimum number of additional fixed traps that any extension of path P must pass through to reach the goal. Formally,

$$\mathrm{LB}(P) = \min_{P' \in \mathrm{Ext}(P)} |\mathrm{Forbid}(P') \setminus \mathrm{Forbid}(P)|.$$

Then, $(1-p)^{|\mathrm{Forbid}(P)+\mathrm{LB}(P)|}$ is the least upper bound of $\Pr(\{P'\})$ for all $P' \in \mathrm{Ext}(P)$.

Note that computing $\mathrm{LB}(P)$ exactly is the same as solving kSP under the uniform fixed trap model with $n = k = 1$, which is a NP-hard problem by Theorem 5.1. In order to get an upper bound of all $\Pr(\{P'\})$, where P' is in $\mathrm{Ext}(P)$, it suffices to

Algorithm 3: Heuristic algorithm for kSP

```
input : $(G, M, \{(s_i, g_i)\}_{i=1}^n, k, w, T)$, where $(G, M, \{(s_i, g_i)\}_{i=1}^n, k)$ is an
        instance of $k$SP, $w$ is a parameter of the path generation, and
        $T$ is a parameter of the path improvement.
output: $n$ paths connecting $(s_i, g_i)$ respectively.
$\mathcal{R}$ = path_generation($w$)
$\mathcal{S}$ = path_selection($\mathcal{R}$)
$\mathcal{S}$ = path_replacement($\mathcal{S}, \mathcal{R}$)
$\mathbf{Q} = \emptyset$
while $|\mathbf{Q}| < T$ do
    $\mathcal{S}$ = path_shortening($\mathcal{S}$)
    $\mathbf{Q} = \mathbf{Q} \cup \{\mathcal{S}\}$
    $\mathcal{S}$ = escape($\mathcal{S}$)
return the best solution in $\mathbf{Q}$.
```

obtain a lower bound of $\mathrm{LB}(P)$.

Our idea of obtaining a lower bound of $\mathrm{LB}(P)$ is as follows. Let F_i be a fixed trap that i does not belong to $\mathrm{Forbid}(P)$. If an extension P' of P passes through one vertex of F_i, then charge P' by $1/|F_i \cap V|$. Thus, if an extension P' of P passes through one $v \in V$, then we charge P' by $\sum_{i \notin \mathrm{Forbid}(P), v \in F_i} 1/|F_i \cap V|$. The minimum charge of any extension of P that reaches the goal, $\mathrm{LB}'(P)$, can be computed efficiently by using a shortest path algorithm.

It is easy to see that $\mathrm{LB}'(P)$ is a lower bound of $\mathrm{LB}(P)$ and we know

$$\mathrm{LB}'(P) \leq \mathrm{LB}(P) \leq \max_i |F_i| \cdot \mathrm{LB}'(P).$$

Thus, we can use $\hat{h}(P) = (1-p)^{|\mathrm{Forbid}(P) + \mathrm{LB}'(P)|}$ to obtain a heuristic function h.

5.5.2 Heuristic algorithm

The previous algorithm uses a heuristic function for pruning, but is guaranteed to find optimal solutions. The heuristic algorithm described in this section does not provide this guarantee. There are three phases: path generation, path selection, and path improvement. Due to the high-dimensional search space of kSP, we first generate a set of candidate paths with size $w \gg n$ to reduce the search space to these w paths. Then, we heuristically find n paths among the set of candidate paths as an initial solution. Finally, we use local search to improve the solution until the process is stabilized. Algorithm 3 outlines the approach.

Since computation of k-survivability is potentially expensive, we only use the computation of k-survivability in the last phase. Moreover, this heuristic algorithm only needs a black box to compute k-survivability, and the same algorithm can be used for both fixed trap and variable trap models.

Path generation

The purpose of this phase is to generate a set $\mathcal{R}$ of $w \gg n$ paths. We design two methods: random generation, and an iterative penalty approach.

Random generation method. To generate one random path, we generate a random spanning tree first and then pick the unique path between s and g on the tree. We repeat this process until w paths are generated.

Iterative penalty method. Another way to generate w paths is repeatedly apply a shortest path algorithm. After a shortest path P is found, we increase the edge weights of all edges in P and repeat. Akgün *et al.* discuss several variants of iterative penalty methods that have different ways to penalize the path (Akgün, Erkut, and Batta 2000).

Path selection

The purpose of this phase is to generate a set n paths among w candidate paths generated in the path generation phase. Although we can design an algorithm to find n paths that maximize k-survivability, since the computation of k-survivability is exponential in n, this approach would be expensive. Thus, our strategy is to use different heuristics to obtain an initial solution without evaluating k-survivability. Then, improve the initial solution based on k-survivability in the next phase.

We find an initial solution by solving a different but related optimization problem.

Distance-based heuristic. We use $d_G(P, P')$ to denote the distance between two paths P and P' on a graph G. One candidate of the distance function is discrete Fréchet distance (Voss, Moll, and Kavraki 2015) and other candidates of distance function can be found in Knepper's thesis (Knepper 2011).

Based on the distance function, we can set up several optimization problems.

(a) remote-clique problem: find

$$\mathcal{S} = \arg \max_{\mathcal{S} \subseteq \mathcal{R}, |\mathcal{S}|=n} \sum_{P,P' \in \mathcal{S}} d_G(P, P').$$

(b) remote-edge problem: find

$$\mathcal{S} = \arg \max_{\mathcal{S} \subseteq \mathcal{R}, |\mathcal{S}|=n} \min_{P,P' \in \mathcal{S}, P \neq P'} d_G(P, P').$$

(c) remote-pseudoforest problem: find

$$\mathcal{S} = \arg \max_{\mathcal{S} \subseteq \mathcal{R}, |\mathcal{S}|=n} \sum_{P \in \mathcal{S}} \min_{P' \in \mathcal{S}, P \neq P'} d_G(P, P').$$

The remote-edge problem is sensitive to the closest-pair of paths, since two solutions with the same closest pair of paths will have the same minimum distance, even if one solution is much longer than the other (Voss, Moll, and Kavraki 2015). Since all these maximum diversity problems are NP-hard, we use heuristic methods to find a good solution (Martí et al. 2013).

Survivability-based heuristic. We also can use Erickson's and LaValle's notion of survivability (Erickson and LaValle 2009) in our heuristic. We heuristically find n paths with high survivability and use this set as an initial solution.

Path improvement

The purpose of this phase is to improve k-survivability of an initial solution $\mathcal{S}$ by using local operations: *path replacement* and *path shortening*. Path replacement iteratively replaces one path to improve k-survivability. Path shortening iteratively replaces a subpath of one path to improve k-survivability.

We first apply path replacement to improve k-survivability and then apply path shortening. Since path improvement is a local search method, the search process may be trapped in a local maximum. Thus, when the search reaches a local maximum, we use a randomized method to escape from the local maximum and then apply path shortening again.

Path replacement. Replace one path in the current solution by another path in $\mathcal{R}$ giving the maximum k-survivability for the set; repeat until no further improvement can be made.

Path shortening. Find the maximum improvement of k-survivability that can be made by replacing one subpath of a path in the current solution by a shortest path on G connecting the endpoints of the subpath. Repeat shortening until no further improvement can be made.

Although path shortening is very effective under the fixed r-disc trap model, path shortening may not be useful in general models. Moreover, for kSP with $k > 1$, shortening just one path at a time may lead to getting trapped in local maxima easily. For example, Figure 5.6c shows such a case; all four overlapping paths would need to be shortened simultaneously and in the same way to allow the four robots to follow a better route.

Escape from local maxima Since path shortening is a local search method, path shortening may get trapped by local maxima. When no path shortening can be made, we randomly pick a path in the current solution and reroute a subpath randomly. Then, run the path shortening method again to reach another local maximum. We repeat this process until T local maxima are found for a threshold value T, and choose the best.

5.6 Experimental results

In this section, we describe several experiments (in simulation) on different heuristic methods, and compare them in terms of computation time and k-survivability. Remember that our heuristic method consists of three phases. We suggest two choices in the path generation phase: random generation (RG) and iterative penalty (IP) methods. We suggest four choices in the path selection phase: remote-clique (RC), remote-edge (RE), remote-pseudoforest (RF), and survivability (SU). Finally, we test two additional methods in the path selection phase:

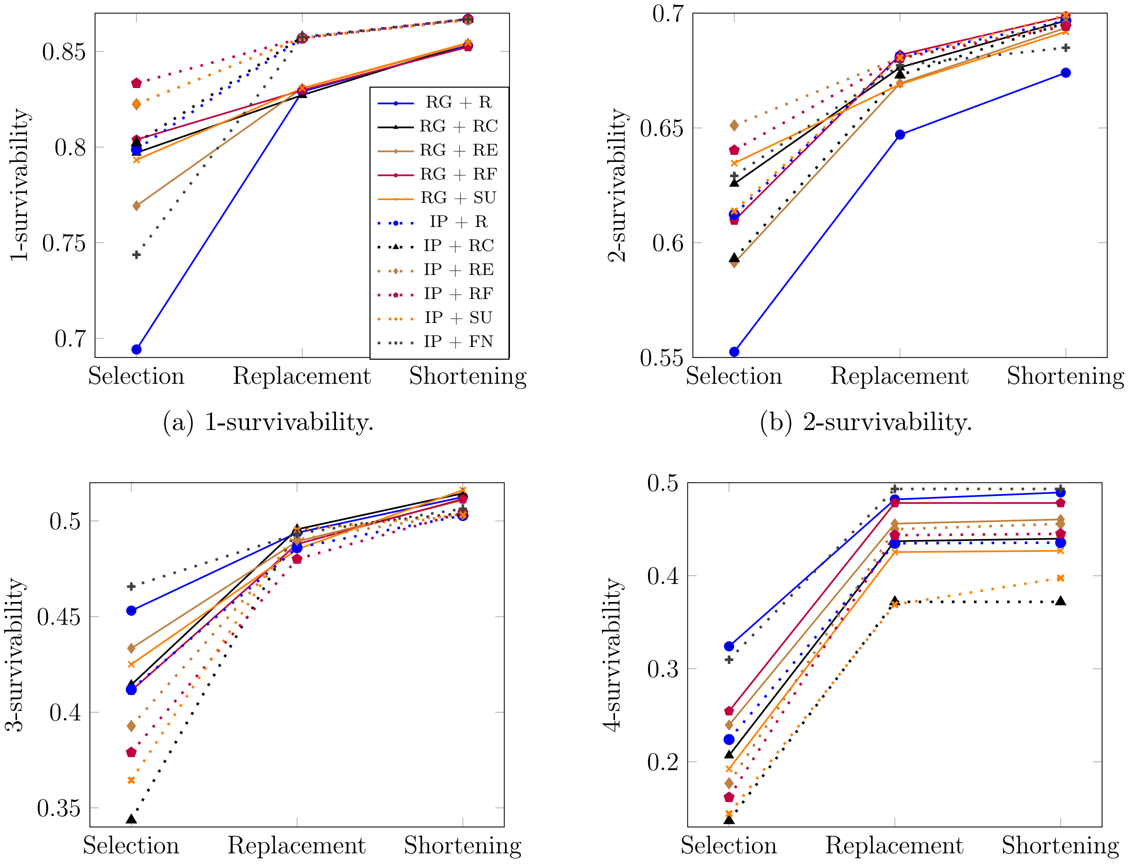
(a) random (R): pick n paths in $\mathcal{R}$ uniformly at random.

(b) first n paths (FN): if the paths are generated by the iterative penalty method, we pick the first n generated paths.

5.6.1 Experiment setup

We used an environment containing 2500 vertices and 80 rectangular obstacles under the fixed r-disc trap model, where $r = 5$ and $p = 0.002$. The environment is shown in Figure 5.6. We used the heuristic algorithm to find $n = 5$ paths with high k-survivability, for $k = 1 \ldots 4$. We generated $w = 50$ paths in the path generation phase and found $T = 3$ local maxima in the path improvement phase.

The heuristic algorithm is implemented in Java and all tests were conducted on a laptop (2010 MacBook Pro) with an Intel Core i5 2.4 GHz CPU and 8GB RAM. We repeated the experiments ten times and took the average of the results.



(a) 1-survivability.

(b) 2-survivability.

(c) 3-survivability.

(d) 4-survivability.

Figure 5.7: All methods' k-survivability at the end of path selection phase, path improvement phase and path shortening phase. All methods that are using random generation method are represented by solid lines and all methods that are using iterative penalty method are represented by dotted lines. Different methods in the second phase are represented by different colors.

5.6.2 Results

We first show k-survivability of each phase for each method in Figure 5.7. When k is small, path shortening effectively improves the k-survivability and the iterative penalty method tends to perform better. However, when $k = 4$, path shortening is not effective, since our algorithm only tries to shorten one path at a time but escaping

Path generation method	Path selection method	Path generation	Path selection	Path replacement	Path improvement	Total time
Random generation method	Random	37.3	**0.0**	2124.3	17360.9	19533.3
	Remote-clique	31.0	2894.6	**1577.1**	17329.0	21841
	Remote-edge	29.3	2505.0	1788.5	16791.9	21125.8
	Remote-pseudoforect	28.6	2649.5	1854.2	16441.6	20986.7
	Survivability	**27.2**	48686.9	1613.3	18391.3	68733.7
Iterative penalty method	Random	430.0	**0.0**	2475.7	**15752.9**	**18676.3**
	Remote-clique	381.9	4344.8	3300.1	18795.2	26841.3
	Remote-edge	384.7	3550.3	3017.6	17998.5	24972.5
	Remote-pseudoforest	379.9	4112.4	2673.5	18782.7	25970.8
	Survivability	380.0	56769.0	1991.2	15978.2	75144.8
	First-n	375.3	**0.0**	2968.9	18934.4	22305.3

Table 5.1: Running times measured in milliseconds for $k = 1$.

Path generation method	Path selection method	Path generation	Path selection	Path replacement	Path improvement	Total time
Random generation method	Random	32.5	**0.0**	**82467.6**	164975.6	247629
	Remote-clique	32.6	2996.8	141197.7	161304.4	305704.5
	Remote-edge	**30.4**	2501.5	94436.1	173701.5	270834.6
	Remote-pseudoforect	31.7	2541.7	120313.1	148490.6	271552.4
	Survivability	30.9	43867.7	92570.7	173222.5	309874.5
Iterative penalty method	Random	450.6	**0.0**	157176.9	134173.1	292029.6
	Remote-clique	391.6	4567.0	121918.6	171172.5	298295.6
	Remote-edge	395.1	3942.3	141077.8	137397.3	283055.5
	Remote-pseudoforest	395.3	4289.3	145943.7	148716.3	299595.0
	Survivability	386.1	61717.0	152336.0	139777.2	354452.3
	First-n	401.1	**0.0**	93838.4	**110945.8**	**205379.2**

Table 5.2: Running times measured in milliseconds for $k = 2$.

from a local maxima may require shortening several paths at the same time.

We measured the running time for all methods maximizing k-survivability; the running times for each method for $k = 1 \sim 4$ are shown in Tables 5.1, 5.2, 5.3, and 5.4. When $k = 1$, the most efficient algorithm is IP + R, while IP + FN becomes the most efficient algorithm when $k > 1$. Note that IP + FN is the second fastest method when $k = 1$ and produce high quality results when $k = 1, 3, 4$. This may hint that although kSP is hard in general, kSP under the fixed r-disc trap model may be tractable.

When k increases, since evaluating k-survivability takes more time, the running time for our algorithm is increasing as well. However, when $k = 4$, since the greedy approach is trapped in local maxima easily when k is large, our algorithm takes fewer iterations and the running time decreases dramatically.

Path generation method	Path selection method	Path generation	Path selection	Path replacement	Path improvement	Total time
Random generation method	Random	38.0	**0.0**	91833.0	131298.2	223307.1
	Remote-clique	30.7	2814.6	111247.5	159241.1	273482.3
	Remote-edge	32.1	2341.6	90187.8	147532.2	240240.6
	Remote-pseudoforect	32.8	2546.7	97935.0	157753.6	258433.5
	Survivability	**28.1**	42983.5	101884.9	122851.2	267926.2
Iterative penalty method	Random	478.2	**0.0**	99059.3	129456.2	229169.4
	Remote-clique	402.6	4326.8	147504.6	**86207.7**	238664.7
	Remote-edge	406.3	3932.5	123417.1	108172.1	236117.3
	Remote-pseudoforest	406.6	4068.0	119607.4	120956.7	245250.0
	Survivability	407.3	50149.1	113635.4	99078.2	263541.5
	First-n	408.9	**0.0**	**72034.4**	147989.8	**220590.8**

Table 5.3: Running times measured in milliseconds for $k = 3$.

Path generation method	Path selection method	Path generation	Path selection	Path replacement	Path improvement	Total time
Random generation method	Random	31.8	**0.0**	**1947.1**	9342.4	11331.9
	Remote-clique	31.8	3196.1	2625.1	5852.5	11716.7
	Remote-edge	34.0	2779.2	2397.3	8340.0	13563.0
	Remote-pseudoforect	**30.9**	2849.7	2380.8	6360.0	11635.0
	Survivability	31.8	50428.3	2548.6	7262.6	60287.2
Iterative penalty method	Random	428.6	**0.0**	2410.7	8758.6	11615.5
	Remote-clique	392.7	4428.6	4017.4	6890.7	15751.0
	Remote-edge	395.6	3859.3	3507.8	7006.7	14790.8
	Remote-pseudoforest	391.2	3954.8	3586.8	6146.4	14103.3
	Survivability	388.6	59955.6	3464.7	7223.7	71058.8
	First-n	396.1	**0.0**	2153.5	**5660.9**	**8235.7**

Table 5.4: Running times measured in milliseconds for $k = 4$.

5.7 Conclusion and future work

This work is preliminary, and considers only simple k-survivability problems; however, we believe that k-survivability motivates a wealth of interesting practical and theoretical problems. For example, the problem of k-survivability might be reversed to plan defenses against infiltration or attack. Not all applications of k-survivability need be violent. For example, k-survivability can be considered in the context of visibility or stealth, as has turned out to be central in multi-robot pursuit-evasion games (Chung, Hollinger, and Isler 2011; Murphy 2014; Acar and Choset 2002) for search-and-rescue operations. With a model of feedback or communication, we imagine that k-survivability might also provide some insights into collaboration and cooperation problems such as those that arise in sports (Biswas et al. 2014) or control of large robot swarms (Rubenstein et al. 2014).

Several future directions of theoretical research are possible. Continuous-space models might be approached using variational calculus or optimal control techniques (Zabarankin, Uryasev, and Murphey 2006; Miller et al. 2011). Obstacles such that the risk of a path depends on the distance between the robot and the obstacle, as for paths in mined water (Babel and Zimmermann 2015) are a potential future direction, as are time-dependent obstacles.

5.7.1 A heuristic for escaping from local maxima for $n > k > 1$

We note that when $n > k > 1$, path shortening may not be effective, since Algorithm 3 shortens only one path at a time, but escaping from a local maxima may require shortening several paths at the same time. In particular, in order to escape from a local maxima, simultaneously shortening several paths that share the same sub-path may be necessary. Thus, we would like to design a better heuristic for escaping from local maxima when $n > k > 1$.

Let's consider a small example. Suppose that we want to find paths for four robots, R_1, R_2, R_3, and R_4, with good 2-survivability. We can divide four robots into two groups: $G_1 = \{R_1, R_2\}$ and $G_2 = \{R_3, R_4\}$. Then, we use Algorithm 3 to find two paths, P_1 and P_2, with good 1-survivability for G_1 and G_2 respectively. That is, robots in G_1 follow P_1 and robots in G_2 follow P_2

Since the paths P_1 and P_2 are locally maxima for 1-survivability, each group cannot improve 2-survivability by shortening its path. However, a robot may improve 2-survivability by deviating from its group. Thus, we can apply path shortening again for these four robots, since path shortening only modifies one path at a time.

This idea can be generalized: in order to solve a problem for $n > k > 1$, we

	2-survivability	Total running time
Algorithm 3	0.631770	20736
Algorithm 3 with grouping heuristic	0.698003	19664

Table 5.5: Running times measured in milliseconds for $n = 4$ and $k = 2$.

	2-survivability	Total running time
Algorithm 3	0.684203	279015
Algorithm 3 with grouping heuristic	0.693588	108594

Table 5.6: Running times measured in milliseconds for $n = 5$ and $k = 2$.

firstly find $n' = \lceil \frac{n}{k} \rceil$ paths with good 1-survivability. Then, duplicate each path k or $k - 1$ times to obtain n paths. Finally, use the adjusted solution as the initial solution in the path shortening phase for solving the original $n > k > 1$ problem. We call this heuristic as *grouping heuristic*. In our example, we use two paths with good 1-survivability as an initial solution for finding four paths with good 2-survivability.

We conducted an experiment in the same environment as the environment in the experiment in Section 5.6. For path generation, we used the random generation method. For path selection, we used the random selection method. We first test the heuristic for finding four paths with good 2-survivability. The result is in Table 5.5. The new method obtains paths with higher 2-survivability.

Then, we tested the new method for finding five paths with good 2-survivability. The result is in Table 5.6. The new method obtains paths with higher 2-survivability in less time!

Bibliography

Acar, Ercan U. and Howie Choset. "Sensor-based Coverage of Unknown Environments". In: *International Journal of Robotics Research* 21.4 (April 2002), pages 345–366. ISSN: 0278-3649. DOI: 10.1177/027836402320556368. URL: http://dx.doi.org/10.1177/027836402320556368 (cited on page 140).

Agarwal, Pankaj K. and Hongyan Wang. "Approximation Algorithms for Curvature-Constrained Shortest Paths". In: *SIAM Journal on Computing* 30.6 (2001), pages 1739–1772. ISSN: 0097-5397. DOI: 10.1137/S0097539796307790. URL: http://dx.doi.org/10.1137/S0097539796307790 (cited on page 22).

Agarwal, Pankaj K., Therese C. Biedl, Sylvain Lazard, Steve Robbins, Subhash Suri, and Sue Whitesides. "Curvature-Constrained Shortest Paths in a Convex Polygon". In: *SIAM Journal on Computing* 31.6 (January 2002), pages 1814–1851. ISSN: 0097-5397. DOI: 10.1137/S0097539700374550. URL: http://dx.doi.org/10.1137/S0097539700374550 (cited on page 22).

Akgün, Vedat, Erhan Erkut, and Rajan Batta. "On finding dissimilar paths". In: *European Journal of Operational Research* 121.2 (March 2000), pages 232–246. ISSN: 0377-2217. DOI: 10.1016/S0377-2217(99)00214-3. URL: http://dx.doi.org/10.1016/S0377-2217(99)00214-3 (cited on pages 124, 134).

Aljazzar, Husain and Stefan Leue. "K^*: A heuristic search algorithm for finding the k shortest paths". In: *Artificial Intelligence* 175.18 (December 2011), pages 2129–2154. ISSN: 0004-3702. DOI: 10.1016/j.artint.2011.07.003. URL: http://dx.doi.org/10.1016/j.artint.2011.07.003 (cited on page 122).

Andreas, April K. and J. Cole Smith. "Mathematical Programming Algorithms for Two-Path Routing Problems with Reliability Considerations". In: *INFORMS Journal on Computing* 20.4 (2008), pages 553–564. ISSN: 1091-9856. DOI: 10.1287/ijoc.1080.0266. URL: http://dx.doi.org/10.1287/ijoc.1080.0266 (cited on page 123).

Arslan, Oktay and Panagiotis Tsiotras. "Use of relaxation methods in sampling-based algorithms for optimal motion planning". In: *IEEE International Conference on Robotics and Automation, ICRA*. May 2013, pages 2421–2428. ISBN: 978-1-4673-5641-1. DOI: 10.1109/ICRA.2013.6630906. URL: http://dx.doi.org/10.1109/ICRA.2013.6630906 (cited on pages 2, 10).

Assadi, Sepehr, Ehsan Emamjomeh-Zadeh, Ashkan Norouzi-Fard, Sadra Yazdanbod, and Hamid Zarrabi-Zadeh. "The Minimum Vulnerability Problem". In: *Algorithmica* 70.4 (December 2014), pages 718–731. ISSN: 0178-4617. DOI: 10.1007/s00453-014-9927-z. URL: http://dx.doi.org/10.1007/s00453-014-9927-z (cited on page 123).

Babel, Luitpold and Thomas Zimmermann. "Planning safe navigation routes through mined waters". In: *European Journal of Operational Research* 241.1 (February 2015), pages 99–108. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2014.08.002. URL: http://dx.doi.org/10.1016/j.ejor.2014.08.002 (cited on pages 3, 122, 141).

Balkcom, Devin J., Paritosh A. Kavathekar, and Matthew T. Mason. "Time-optimal Trajectories for an Omni-directional Vehicle". In: *International Journal of Robotics Research* 25.10 (October 2006), pages 985–999. ISSN: 0278-3649. DOI: 10.1177/0278364906069166. URL: http://dx.doi.org/10.1177/0278364906069166 (cited on pages 2, 4, 23).

Balkcom, Devin J. and Matthew T. Mason. "Extremal Trajectories for Bounded Velocity Differential Drive Robots". In: *IEEE International Conference on Robotics and Automation, ICRA*. April 2000, pages 2479–2484. DOI: 10.1109/ROBOT.2000.846401. URL: http://dx.doi.org/10.1109/ROBOT.2000.846401 (cited on page 23).

Balkcom, Devin J. and Matthew T. Mason. "Extremal Trajectories for Bounded Velocity Mobile Robots". In: *IEEE International Conference on Robotics and Automation, ICRA*. May 2002, pages 1747–1752. ISBN: 0-7803-7272-7. DOI: 10.1109/ROBOT.2002.1014794. URL: http://dx.doi.org/10.1109/ROBOT.2002.1014794 (cited on page 23).

Balkcom, Devin J. and Matthew T. Mason. "Geometric Construction of Time Optimal Trajectories for Differential Drive Robots". In: *Workshop on the Algorithmic Foundations of Robotics, WAFR*. 2000. URL: http://www.cs.dartmouth.edu/~devin/papers/djb-wafr00.pdf (cited on page 23).

Balkcom, Devin J. and Matthew T. Mason. "Time Optimal Trajectories for Bounded Velocity Differential Drive Robots". In: *IEEE International Conference on Robotics and Automation, ICRA*. April 2000, pages 2499–2504. DOI: 10.1109/ROBOT.2000.846404. URL: http://dx.doi.org/10.1109/ROBOT.2000.846404 (cited on page 23).

Balkcom, Devin J. and Matthew T. Mason. "Time Optimal Trajectories for Bounded Velocity Differential Drive Vehicles". In: *International Journal of Robotics Research* 21.3 (March 2002), pages 199–218. ISSN: 0278-3649. DOI: 10.1177/027836402320556403. URL: http://dx.doi.org/10.1177/027836402320556403 (cited on pages 2, 4, 23).

Balkcom, Devin J., Ajay Kannan, Yu-Han Lyu, Weifu Wang, and Yinan Zhang. "Metric cells: Towards complete search for optimal trajectories". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. September 2015, pages 4941–4948. DOI: 10.1109/IROS.2015.7354072. URL: http://dx.doi.org/10.1109/IROS.2015.7354072 (cited on page 5).

Bardi, Martino and Italo Capuzzo-Dolcetta. *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*. Modern Birkhäuser Classics. Birkhäuser, January 1997. ISBN: 978-0-81764-754-4. DOI: 10.1007/978-0-8176-4755-1 (cited on page 21).

Barraquand, Jérîme and Jean-Claude Latombe. "Robot motion planning: a distributed representation approach". In: *International Journal of Robotics Research* 10.6 (December 1991), pages 628–649. ISSN: 0278-3649. DOI: 10.1177/027836499101000604 (cited on page 16).

Barraquand, Jérôme, Bruno Langlois, and Jean-Claude Latombe. "Numerical potential field techniques for robot path planning". In: *IEEE Transactions on Systems, Man, and Cybernetics* 22.2 (March 1992), pages 224–241. ISSN: 0018-9472. DOI: 10.1109/21.148426. URL: http://dx.doi.org/10.1109/21.148426 (cited on page 9).

Barraquand, Jérôme and Jean-Claude Latombe. "Nonholonomic Multibody Mobile Robots: Controllability and Motion Planning in the Presence of Obstacles". In: *Algorithmica* 10.2-4 (October 1993), pages 121–155. ISSN: 0178-4617. DOI: 10.1007/BF01891837. URL: http://dx.doi.org/10.1007/BF01891837 (cited on pages 12, 89).

Barraquand, Jérôme, Lydia E. Kavraki, Jean-Claude Latombe, Tsai-Yen Li, Rajeev Motwani, and Prabhakar Raghavan. "A Random Sampling Scheme for Path Planning". In: *International Journal of Robotics Research* 16.6 (December 1997), pages 759–774. ISSN: 0278-3649. DOI: 10.1177/027836499701600604. URL: http://dx.doi.org/10.1177/027836499701600604 (cited on page 10).

Basu, Saugata, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry*. 2nd edition. Volume 10. Algorithms and Computation in Mathematics. Springer-Verlag Berlin Heidelberg, August 2006. ISBN: 978-3-540-33098-1. DOI: 10.1007/3-540-33099-2. URL: http://www.springer.com/us/book/9783540330981 (cited on page 7).

Basu, Saugata, Richard Pollack, and Marie-Françoise Roy. "Computing roadmaps of semi-algebraic sets on a variety". In: *Journal of the American Mathematical Society* 13.1 (January 2000), pages 55–82. ISSN: 0894-0347. DOI: 10.1090/S0894-0347-99-00311-2. URL: http://www.ams.org/journals/jams/2000-13-01/S0894-0347-99-00311-2/ (cited on page 7).

Basu, Saugata and Marie-Françoise Roy. "Divide and Conquer Roadmap for Algebraic Sets". In: *Discrete & Computational Geometry* 52.2 (September 2014), pages 278–343. ISSN: 0179-5376. DOI: 10.1007/s00454-014-9610-9. URL: http://dx.doi.org/10.1007/s00454-014-9610-9 (cited on page 7).

Basu, Saugata, Marie-Françoise Roy, Mohab Safey El Din, and Éric Schost. “A Baby Step-Giant Step Roadmap Algorithm for General Algebraic Sets”. In: *Foundations of Computational Mathematics* 14.6 (October 2014), pages 1117–1172. ISSN: 1615-3375. DOI: 10.1007/s10208-014-9212-1. URL: http://dx.doi.org/10.1007/s10208-014-9212-1 (cited on page 7).

Bazaraa, Mokhtar S., Hanif D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. 3rd edition. Wiley-Interscience, May 2006. ISBN: 978-0-47148-600-8. URL: http://www.wiley.com/WileyCDA/WileyTitle/productCd-0471486000.html (cited on pages 27, 29).

Bellaïche, André and Jean-Jacques Risler, editors. *Sub-Riemannian Geometry*. Volume 144. Progress in Mathematics. Birkhäuser Basel, October 1996. ISBN: 978-3-03489-946-8. DOI: 10.1007/978-3-0348-9210-0 (cited on page 12).

Bellman, Richard E. *Dynamic Programming*. Princeton Landmarks in Mathematics and Physics. Princeton University Express, July 2010. ISBN: 978-0-69114-668-3. URL: http://press.princeton.edu/titles/9234.html (cited on page 20).

Bentley, Jon Louis and Andrew Chi-Chih Yao. “An Almost Optimal Algorithm for Unbounded Searching”. In: *Information Processing Letters* 5.3 (August 1976), pages 82–87. ISSN: 0020-0190. DOI: 10.1016/0020-0190(76)90071-5. URL: http://dx.doi.org/10.1016/0020-0190(76)90071-5 (cited on page 107).

Berkovitz, Leonard David and Negash G. Medhin. *Nonlinear Optimal Control Theory*. Applied Mathematics & Nonlinear Science. Chapman & Hall/CRC, August 2012. ISBN: 978-1-46656-026-0. URL: http://www.crcpress.com/product/isbn/9781466560260 (cited on page 13).

Bespamyatnikh, Sergei. “Computing homotopic shortest paths in the plane”. In: *Journal of Algorithms* 49.2 (November 2003), pages 284–303. ISSN: 0196-6774. DOI: 10.1016/S0196-6774(03)00090-7. URL: http://dx.doi.org/10.1016/S0196-6774(03)00090-7 (cited on page 122).

Bhandari, Ramesh. *Survivable Networks: Algorithms for Diverse Routing*. Volume 477. The Springer International Series in Engineering and Computer Science. Springer, 1999. ISBN: 978-0-7923-8381-9. URL: http://www.springer.com/us/book/9780792383819 (cited on page 123).

Bhatia, Randeep, Murali S. Kodialam, and T. V. Lakshman. “Finding disjoint paths with related path costs”. In: *Journal of Combinatorial Optimization* 12.1-2 (September 2006), pages 83–96. ISSN: 1382-6905. DOI: 10.1007/s10878-006-8906-y. URL: http://dx.doi.org/10.1007/s10878-006-8906-y (cited on page 123).

Bhattacharya, Subhrajit, Maxim Likhachev, and Vijay Kumar. “Topological constraints in search-based robot path planning”. In: *Autonomous Robots* 33.3 (October 2012), pages 273–290. ISSN: 1573-7527. DOI: 10.1007/s10514-012-9304-1. URL: http://dx.doi.org/10.1007/s10514-012-9304-1 (cited on pages 11, 122).

Bhattacharya, Subhrajit, David Lipsky, Robert Ghrist, and Vijay Kumar. “Invariants for homology classes with application to optimal search and planning problem in robotics”. In: *Annals of Mathematics and Artificial Intelligence* 67.3-4 (March 2013), pages 251–281. ISSN: 1012-2443. DOI: 10.1007/s10472-013-9357-7 (cited on pages 11, 113).

Bialkowski, Joshua, Sertac Karaman, Michael W. Otte, and Emilio Frazzoli. “Efficient Collision Checking in Sampling-Based Motion Planning”. In: *Workshop* *on the Algorithmic Foundations of Robotics, WAFR*. Volume 86. Springer Tracts in Advanced Robotics. June 2012, pages 365–380. ISBN: 978-3-642-36278-1. DOI: 10.1007/978-3-642-36279-8_22. URL: http://dx.doi.org/10.1007/978-3-642-36279-8_22 (cited on page 93).

Biswas, Joydeep, Juan Pablo Mendoza, Danny Zhu, Benjamin Choi, Steven D. Klee, and Manuela M. Veloso. "Opponent-driven planning and execution for pass, attack, and defense in a multi-robot soccer team". In: *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS*. May 2014, pages 493–500. ISBN: 978-1-4503-2738-1. URL: http://dl.acm.org/citation.cfm?id=2615812 (cited on page 140).

Blatt, John M. "Optimal control with a cost of switching control". In: *Journal of the Australian Mathematical Society*. B. Applied Mathematics 19 (03 June 1976), pages 316–332. ISSN: 1446-8735. DOI: 10.1017/S0334270000001181. URL: http://journals.cambridge.org/article_S0334270000001181 (cited on pages 17, 26, 27).

Bloch, Anthony, J. Baillieul, P. Crouch, and Jerrold Marsden. *Nonholonomic Mechanics and Control*. Volume 24. Interdisciplinary Applied Mathematics. Springer, September 2007. ISBN: 978-0-38795-535-3. DOI: 10.1007/b97376 (cited on page 11).

Boidot, Emmanuel, Aude Marzuoli, and Eric Feron. "Optimal Planning Strategy for Ambush Avoidance". In: *Workshop of Artificial Intelligence for Transportation: Advice, Interactivity and Actor Modeling at Conference on Artificial Intelligence AAAI*. January 2015. URL: http://www.aaai.org/ocs/index.php/WS/AAAIW15/paper/view/10194/10153 (cited on pages 3, 122).

Boissonnat, Jean-Daniel, André Cérézo, and Juliette Leblond. “Shortest paths of bounded curvature in the plane”. In: *Journal of Intelligent and Robotic Systems* 11.1-2 (March 1994), pages 5–20. ISSN: 0921-0296. DOI: 10.1007/BF01258291. URL: http://dx.doi.org/10.1007/BF01258291 (cited on page 23).

Boissonnat, Jean-Daniel and Sylvain Lazard. “A Polynomial-Time Algorithm for Computing Shortest Paths of Bounded Curvature AmidstModerate Obstacles”. In: *International Journal of Computational Geometry & Applications* 13.3 (June 2003), pages 189–229. ISSN: 0218-1959. DOI: 10.1142/S0218195903001128. URL: http://dx.doi.org/10.1142/S0218195903001128 (cited on page 22).

Bonnard, Bernard and Monique Chyba. *Singular Trajectories and their Role in Control Theory*. Volume 40. Mathématiques et Applications. Springer, May 2003. ISBN: 978-3-540-00838-5. URL: http://www.springer.com/gp/book/9783540008385 (cited on page 25).

Boscain, Ugo and Benedetto Piccoli. *Optimal Syntheses for Control Systems on 2-D Manifolds*. Volume 43. Mathématiques et Applications. Springer, January 2004. ISBN: 978-3-540-20306-3. URL: http://www.springer.com/gp/book/9783540203063 (cited on page 24).

Botea, Adi, Bruno Bouzy, Michael Buro, Christian Bauckhage, and Dana S. Nau. “Pathfinding in Games”. In: *Artificial and Computational Intelligence in Games*. 2013, pages 21–31. ISBN: 978-3-939897-62-0. DOI: 10.4230/DFU.Vol6.12191.21. URL: http://dx.doi.org/10.4230/DFU.Vol6.12191.21 (cited on page 4).

Branicky, Michael S., Ross A. Knepper, and James J. Kuffner. “Path and trajectory diversity: Theory and algorithms”. In: *IEEE International Conference on Robotics and Automation, ICRA*. May 2008, pages 1359–1364. DOI: 10.1109/ROBOT.2008.

4543392. URL: http://dx.doi.org/10.1109/ROBOT.2008.4543392 (cited on pages 115, 121).

Brooks, Rodney A. and Tomás Lozano-Pérez. “A subdivision algorithm in configuration space for findpath with rotation”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 15.2 (May 1985), pages 224–233. ISSN: 0018-9472. DOI: 10.1109/TSMC.1985.6313352. URL: http://dx.doi.org/10.1109/TSMC.1985.6313352 (cited on page 9).

Brumbaugh-Smith, James P. and Douglas R. Shier. “Minimax Models for Diverse Routing”. In: *INFORMS Journal on Computing* 14.1 (February 2002), pages 81–95. ISSN: 1091-9856. DOI: 10.1287/ijoc.14.1.81.7707. URL: http://dx.doi.org/10.1287/ijoc.14.1.81.7707 (cited on page 123).

Bui, Xuân-Nam, Philippe Souères, Jean-Daniel Boissonnat, and Jean-Paul Laumond. “Shortest Path Synthesis for Dubins Non-Holonomic Robot”. In: *IEEE International Conference on Robotics and Automation, ICRA*. Volume 1. May 1994, pages 2–7. ISBN: 0-8186-5330-2. DOI: 10.1109/ROBOT.1994.351019. URL: http://dx.doi.org/10.1109/ROBOT.1994.351019 (cited on page 22).

Bushnell, Linda G., Dawn M. Tilbury, and Shankar Sastry. “Steering Three-Input Nonholonomic Systems: The Fire Truck Example”. In: *International Journal of Robotics Research* 14.4 (August 1995), pages 366–381. ISSN: 0278-3649. DOI: 10.1177/027836499501400405. URL: http://dx.doi.org/10.1177/027836499501400405 (cited on page 12).

Calin, Ovidiu and Der-Chen Chang. *Sub-Riemannian Geometry: General Theory and Examples*. Encyclopedia of Mathematics and its Applications. Cambridge, April 2009. ISBN: 978-0-52189-730-3. URL: http://www.cambridge.org/us/

academic/subjects/mathematics/geometry-and-topology/sub-riemannian-geometry-general-theory-and-examples (cited on page 12).

Canny, John F. *Complexity of Robot Motion Planning*. The ACM Distinguished Dissertation Series. MIT Press, June 1988. ISBN: 978-0-26203-136-3. URL: http://mitpress.mit.edu/books/complexity-robot-motion-planning (cited on page 7).

Canny, John F. "Computing Roadmaps of General Semi-Algebraic Sets". In: *Computer Journal* 36.5 (1993), pages 504–514. ISSN: 0010-4620. DOI: 10.1093/comjnl/36.5.504. URL: http://dx.doi.org/10.1093/comjnl/36.5.504 (cited on page 7).

Canny, John F., Ashutosh Rege, and John H. Reif. "An Exact Algorithm for Kinodynamic Planning in the Plane". In: *Discrete & Computational Geometry* 6 (1991), pages 461–484. ISSN: 0179-5376. DOI: 10.1007/BF02574702. URL: http://dx.doi.org/10.1007/BF02574702 (cited on page 7).

Chan, David Yu Cheng and David G. Kirkpatrick. "Multi-Path Algorithms for minimum-colour path problems with applications to approximating barrier resilience". In: *Theoretical Computer Science* 553 (October 2014), pages 74–90. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2014.04.009. URL: http://dx.doi.org/10.1016/j.tcs.2014.04.009 (cited on page 128).

Chandra, Barun and Magnús M. Halldórsson. "Approximation Algorithms for Dispersion Problems". In: *Journal of Algorithms* 38.2 (February 2001), pages 438–465. ISSN: 0196-6774. DOI: 10.1006/jagm.2000.1145. URL: http://dx.doi.org/10.1006/jagm.2000.1145 (cited on page 120).

Chazelle, Bernard, Herbert Edelsbrunner, Leonidas J. Guibas, and Micha Sharir. "A Singly Exponential Stratification Scheme for Real Semi-Algebraic Varieties and its

Applications". In: *Theoretical Computer Science* 84.1 (July 1991), pages 77–105. ISSN: 0304-3975. DOI: 10.1016/0304-3975(91)90261-Y. URL: http://dx.doi.org/10.1016/0304-3975(91)90261-Y (cited on page 7).

Chen, Pang C. and Yong K. Hwang. "SANDROS: a dynamic graph search algorithm for motion planning". In: *IEEE Transactions on Robotics and Automation* 14.3 (June 1998), pages 390–403. ISSN: 1042-296X. DOI: 10.1109/70.678449. URL: http://dx.doi.org/10.1109/70.678449 (cited on page 9).

Chitsaz, Hamidreza. "Geodesic problems for mobile robots". PhD thesis. University of Illinois at Urbana-Champaign, 2008. URL: http://hdl.handle.net/2142/11418 (cited on page 23).

Chitsaz, Hamidreza, Steven M. LaValle, Devin J. Balkcom, and Matthew T. Mason. "Minimum Wheel-Rotation Paths for Differential-Drive Mobile Robots". In: *International Journal of Robotics Research* 28.1 (January 2009), pages 66–80. ISSN: 0278-3649. DOI: 10.1177/0278364908096750 (cited on page 23).

Chung, Timothy H., Geoffrey A. Hollinger, and Volkan Isler. "Search and pursuit-evasion in mobile robotics - A survey". In: *Autonomous Robots* 31.4 (November 2011), pages 299–316. ISSN: 0929-5593. DOI: 10.1007/s10514-011-9241-4. URL: http://dx.doi.org/10.1007/s10514-011-9241-4 (cited on page 140).

Chyba, Monique and Thomas Haberkorn. "Autonomous Underwater Vehicles: Singular Extremals and Chattering". In: *Proceedings of the 22nd Conference on Systems, Control, Modeling and Optimization*. Volume 202. July 2006, pages 103–113. DOI: 10.1007/0-387-33882-9_10. URL: http://dx.doi.org/10.1007/0-387-33882-9_10 (cited on page 23).

Chyba, Monique and Sepanta Sekhavat. “Time optimal paths for a mobile robot with one trailer”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. October 1999, pages 1669–1674. ISBN: 0-7803-5184-3. DOI: 10.1109/IROS.1999.811718. URL: http://dx.doi.org/10.1109/IROS.1999.811718 (cited on page 23).

Cidon, Israel, Raphael Rom, and Yuval Shavitt. “Analysis of multi-path routing”. In: *IEEE/ACM Transactions on Networking* 7.6 (December 1999), pages 885–896. ISSN: 1063-6692. DOI: 10.1109/90.811453. URL: http://dx.doi.org/10.1109/90.811453 (cited on page 123).

Clarke, Francis. *Functional Analysis, Calculus of Variations and Optimal Control*. Volume 264. Graduate Texts in Mathematics. Springer, February 2013. ISBN: 978-1-4471-481-9. DOI: 10.1007/978-1-4471-4820-3 (cited on page 21).

Cockayne, E. J. and G. W. C. Hall. “Plane Motion of a Particle Subject to Curvature Constraints”. In: *SIAM Journal on Control* 13.1 (1975), pages 197–220. ISSN: 0036-1402. DOI: 10.1137/0313012. URL: http://dx.doi.org/10.1137/0313012 (cited on page 23).

Collins, George E. “Hauptvortrag: Quantifier elimination for real closed fields by cylindrical algebraic decomposition”. In: *Automata Theory and Formal Languages, 2nd GI Conference*. Volume 33. Lecture Notes in Computer Science. May 1975, pages 134–183. ISBN: 978-3-540-07407-6. DOI: 10.1007/3-540-07407-4_17. URL: http://dx.doi.org/10.1007/3-540-07407-4_17 (cited on page 7).

Deits, Robin and Russ Tedrake. “Computing Large Convex Regions of Obstacle-Free Space Through Semidefinite Programming”. In: *Workshop on the Algorithmic Foundations of Robotics, WAFR*. Volume 107. Springer Tracts in Advanced

Robotics. August 2014, pages 109–124. ISBN: 978-3-319-16594-3. DOI: 10.1007/978-3-319-16595-0_7. URL: http://dx.doi.org/10.1007/978-3-319-16595-0_7 (cited on page 93).

Delling, Daniel, Peter Sanders, Dominik Schultes, and Dorothea Wagner. "Engineering Route Planning Algorithms". In: *Algorithmics of Large and Complex Networks - Design, Analysis, and Simulation*. 2009, pages 117–139. ISBN: 978-3-642-02093-3. DOI: 10.1007/978-3-642-02094-0_7. URL: http://dx.doi.org/10.1007/978-3-642-02094-0_7 (cited on page 4).

Dell'Olmo, Paolo, Monica Gentili, and Andrea Scozzari. "On finding dissimilar Pareto-optimal paths". In: *European Journal of Operational Research* 162.1 (April 2005), pages 70–82. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2003.10.033. URL: http://dx.doi.org/10.1016/j.ejor.2003.10.033 (cited on pages 116, 124).

Denny, Jory and Nancy M. Amato. "Toggle PRM: A Coordinated Mapping of C-Free and C-Obstacle in Arbitrary Dimension". In: *Workshop on the Algorithmic Foundations of Robotics, WAFR*. Volume 86. Springer Tracts in Advanced Robotics. June 2012, pages 297–312. ISBN: 978-3-642-36278-1. DOI: 10.1007/978-3-642-36279-8_18. URL: http://dx.doi.org/10.1007/978-3-642-36279-8_18 (cited on page 92).

Desaulniers, Guy. "On shortest paths for a car-like robot maneuvering around obstacles". In: *Robotics and Autonomous Systems* 17.3 (May 1996), pages 139–148. ISSN: 0921-8890. DOI: 10.1016/0921-8890(96)80512-4. URL: http://dx.doi.org/10.1016/0921-8890(96)80512-4 (cited on pages 15, 23, 107).

Desaulniers, Guy and François Soumis. "An efficient algorithm to find a shortest path for a car-like robot". In: *IEEE Transactions on Robotics and Automation* 11.6

(December 1995), pages 819–828. ISSN: 1042-296X. DOI: 10.1109/70.478429. URL: http://dx.doi.org/10.1109/70.478429 (cited on page 23).

Desaulniers, Guy, François Soumis, and Jean-Charles Laurent. “A Shortest Path Algorithm for a Carlike Robot in a Polygonal Environment”. In: *International Journal of Robotics Research* 17.5 (May 1998), pages 512–530. ISSN: 0278-3649. DOI: 10.1177/027836499801700504. URL: http://dx.doi.org/10.1177/027836499801700504 (cited on page 23).

Dmitruk, Andrei Venediktovich and A. M. Kaganovich. “The Hybrid Maximum Principle is a consequence of Pontryagin Maximum Principle”. In: *Systems & Control Letters* 57.11 (November 2008), pages 964–970. ISSN: 0167-6911. DOI: 10.1016/j.sysconle.2008.05.006. URL: http://dx.doi.org/10.1016/j.sysconle.2008.05.006 (cited on page 21).

Dobson, Andrew and Kostas E. Bekris. “Sparse roadmap spanners for asymptotically near-optimal motion planning”. In: *International Journal of Robotics Research* 33.1 (January 2014), pages 18–47. ISSN: 0278-3649. DOI: 10.1177/0278364913498292. URL: http://dx.doi.org/10.1177/0278364913498292 (cited on page 93).

Donald, Bruce Randall and Patrick G. Xavier. “Provably Good Approximation Algorithms for Optimal Kinodynamic Planning: Robots with Decoupled Dynamics Bounds”. In: *Algorithmica* 14.6 (December 1995), pages 443–479. ISSN: 0178-4617. DOI: 10.1007/BF01586636. URL: http://dx.doi.org/10.1007/BF01586636 (cited on pages 89, 92).

Donald, Bruce Randall, Patrick G. Xavier, John F. Canny, and John H. Reif. “Kinodynamic Motion Planning”. In: *Journal of the ACM* 40.5 (November 1993), pages 1048–1066. ISSN: 0004-5411. DOI: 10.1145/174147.174150. URL: http://doi.acm.org/10.1145/174147.174150 (cited on page 92).

Dubins, Lester Eli. “On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents”. In: *American Journal of Mathematics* 79.3 (July 1957), pages 497–516. ISSN: 0002-9327. URL: http://www.jstor.org/stable/2372560 (cited on pages 2, 4, 15, 22).

Efrat, Alon, Stephen G. Kobourov, and Anna Lubiw. “Computing homotopic shortest paths efficiently”. In: *Computational Geometry* 35.3 (October 2006), pages 162–172. ISSN: 0925-7721. DOI: 10.1016/j.comgeo.2006.03.003. URL: http://dx.doi.org/10.1016/j.comgeo.2006.03.003 (cited on page 122).

Eilam-Tzoreff, Tali. “The Disjoint Shortest Paths Problem”. In: *Discrete Applied Mathematics* 85.2 (June 1998), pages 113–138. ISSN: 0166-218X. DOI: 10.1016/S0166-218X(97)00121-2. URL: http://dx.doi.org/10.1016/S0166-218X(97)00121-2 (cited on page 123).

Eppstein, David. “Finding the k Shortest Paths”. In: *SIAM Journal on Computing* 28.2 (1998), pages 652–673. ISSN: 0097-5397. DOI: 10.1137/S0097539795290477. URL: http://dx.doi.org/10.1137/S0097539795290477 (cited on page 122).

Erickson, Lawrence H. and Steven M. LaValle. “Survivability: Measuring and ensuring path diversity”. In: *IEEE International Conference on Robotics and Automation, ICRA*. May 2009, pages 2068–2073. DOI: 10.1109/ROBOT.2009.5152773. URL: http://dx.doi.org/10.1109/ROBOT.2009.5152773 (cited on pages 5, 121, 135).

Eriksson-Bique, Sylvester David, John Hershberger, Valentin Polishchuk, Bettina Speckmann, Subhash Suri, Topi Talvitie, Kevin Verbeek, and Hakan Yildiz. "Geometric k Shortest Paths". In: *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*. January 2015, pages 1616–1625. ISBN: 978-1-61197-374-7. DOI: 10.1137/1.9781611973730.107. URL: http://dx.doi.org/10.1137/1.9781611973730.107 (cited on page 122).

Evans, Lawrence C. *Partial Differential Equations*. 2nd edition. Volume 19. Graduate Studies in Mathematics. American Mathematical Society, March 2010. ISBN: 978-0-82184-974-3. URL: http://www.ams.org/bookstore-getitem/item=GSM-19-R (cited on page 20).

Farber, Michael. *Invitation to Topological Robotics*. Volume 8. Zurich Lectures in Advanced Mathematics. European Mathematical Society, August 2008. ISBN: 978-3-03719-054-8. DOI: 10.4171/054. URL: http://www.ems-ph.org/books/book.php?proj_nr=78&srch=searchterm|farber (cited on page 11).

Farber, Michael. "Topological Complexity of Motion Planning". In: *Discrete & Computational Geometry* 29.2 (January 2003), pages 211–221. DOI: 10.1007/s00454-002-0760-9. URL: http://dx.doi.org/10.1007/s00454-002-0760-9 (cited on page 11).

Farber, Michael and Mark Grant. "Robot Motion Planning, Weights of Cohomology Classes, and Cohomology Operations". In: *Proceedings of the American Mathematical Society* 136.9 (September 2008), pp. 3339–3349. ISSN: 0002-9939. DOI: 10.1090/S0002-9939-08-09529-4 (cited on page 11).

Fortune, Steven, John E. Hopcroft, and James Wyllie. "The Directed Subgraph Homeomorphism Problem". In: *Theoretical Computer Science* 10 (February 1980), pages 111–

121. ISSN: 0304-3975. DOI: 10.1016/0304-3975(80)90009-2. URL: http://dx.doi.org/10.1016/0304-3975(80)90009-2 (cited on page 123).

Frieder, Asaf and Liam Roditty. “An Experimental Study on Approximating k Shortest Simple Paths”. In: *ACM Journal of Experimental Algorithmics* 19.1 (February 2014). ISSN: 1084-6654. DOI: 10.1145/2630068. URL: http://doi.acm.org/10.1145/2630068 (cited on page 122).

Furtuna, Andrei A. “Minimum Time Kinematic Trajectories for Self-Propelled Rigid Bodies in the Unobstructed Plane”. PhD thesis. Dartmouth College, June 2011. URL: http://www.cs.dartmouth.edu/~devin/papers/TR2011-694.pdf (cited on pages 3, 17, 18, 23, 25, 31, 32, 34–37, 44, 47, 67, 68, 72, 74).

Furtuna, Andrei A. and Devin J. Balkcom. “Generalizing Dubins Curves: Minimum-time Sequences of Body-fixed Rotations and Translations in the Plane”. In: *International Journal of Robotics Research* 29.6 (May 2010), pages 703–726. ISSN: 0278-3649. DOI: 10.1177/0278364910365093. URL: http://dx.doi.org/10.1177/0278364910365093 (cited on pages 3, 23, 25).

Furtuna, Andrei A., Wenyu Lu, Weifu Wang, and Devin J. Balkcom. “Minimum-time trajectories for kinematic mobile robots and other planar rigid bodies with finite control sets”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. September 2011, pages 4321–4328. DOI: 10.1109/IROS.2011.6094923. URL: http://dx.doi.org/10.1109/IROS.2011.6094923 (cited on pages 3, 23, 25).

Furtuna, Andrei A., Weifu Wang, Yu-Han Lyu, and Devin J. Balkcom. “Structure and geometry of minimum-time trajectories for planar rigid bodies”. In: *51st Annual Allerton Conference on Communication, Control, and Computing*. October

2013, pages 1584–1591. ISBN: 978-1-4799-3409-6. DOI: 10.1109/Allerton.2013.6736717. URL: http://dx.doi.org/10.1109/Allerton.2013.6736717 (cited on page 3).

Garavello, Mauro and Benedetto Piccoli. “Hybrid Necessary Principle”. In: *SIAM Journal Control and Optimization* 43.5 (2005), pages 1867–1887. ISSN: 0363-0129. DOI: 10.1137/S0363012903416219. URL: http://dx.doi.org/10.1137/S0363012903416219 (cited on page 21).

Giordano, Paolo Robuffo and Marilena Vendittelli. “Shortest Paths to Obstacles for a Polygonal Dubins Car”. In: *IEEE Transactions on Robotics* 25.5 (October 2009), pages 1184–1191. ISSN: 1552-3098. DOI: 10.1109/TRO.2008.2011421. URL: http://dx.doi.org/10.1109/TRO.2008.2011421 (cited on page 23).

Giordano, Paolo Robuffo, Marilena Vendittelli, Jean-Paul Laumond, and Philippe Souères. “Nonholonomic distance to polygonal obstacles for a car-like robot of polygonal shape”. In: *IEEE Transactions on Robotics* 22.5 (October 2006), pages 1040–1047. ISSN: 1552-3098. DOI: 10.1109/TRO.2006.878956. URL: http://dx.doi.org/10.1109/TRO.2006.878956 (cited on page 23).

Goaoc, Xavier, Hyo-Sil Kim, and Sylvain Lazard. “Bounded-Curvature Shortest Paths through a Sequence of Points Using Convex Optimization”. In: *SIAM Journal on Computing* 42.2 (2013), pages 662–684. ISSN: 0097-5397. DOI: 10.1137/100816079. URL: http://dx.doi.org/10.1137/100816079 (cited on page 22).

Gomes, Teresa and Mateusz Zotkiewicz. “Resilient arcs and node disjointness in diverse routing”. In: *Optical Switching and Networking* 14 (August 2014), pages 149–163. ISSN: 1573-4277. DOI: 10.1016/j.osn.2014.05.022. URL: http://dx.doi.org/10.1016/j.osn.2014.05.022 (cited on page 124).

Gotthilf, Zvi and Moshe Lewenstein. “Improved algorithms for the k simple shortest paths and the replacement paths problems”. In: *Information Processing Letters* 109.7 (March 2009), pages 352–355. ISSN: 0020-0190. DOI: 10.1016/j.ipl.2008.12.015. URL: http://dx.doi.org/10.1016/j.ipl.2008.12.015 (cited on page 122).

Green, Colin J. and Alonzo Kelly. “Toward Optimal Sampling in the Space of Paths”. In: *Robotics Research*. Edited by Makoto Kaneko and Yoshihiko Nakamura. Volume 66. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2011, pages 281–292. ISBN: 978-3-642-14742-5. DOI: 10.1007/978-3-642-14743-2_24. URL: http://dx.doi.org/10.1007/978-3-642-14743-2_24 (cited on pages 115, 121).

Halldórsson, Magnús M., Kazuo Iwano, Naoki Katoh, and Takeshi Tokuyama. “Finding Subsets Maximizing Minimum Structures”. In: *SIAM Journal on Discrete Mathematics* 12.3 (1999), pages 342–359. ISSN: 0895-4801. DOI: 10.1137/S0895480196309791. URL: http://dx.doi.org/10.1137/S0895480196309791 (cited on page 120).

Halperin, Dan and Micha Sharir. “A Near-Quadratic Algorithm for Planning the Motion of a Polygon in a Polygonal Environment”. In: *Discrete & Computational Geometry* 16.2 (February 1996), pages 121–134. ISSN: 0179-5376. DOI: 10.1007/BF02716803. URL: http://dx.doi.org/10.1007/BF02716803 (cited on page 7).

Hartl, Richard F., Suresh P. Sethi, and Raymond G. Vickson. “A Survey of the Maximum Principles for Optimal Control Problems with State Constraints”. In: *SIAM Review* 37.2 (1995), pages 181–218. ISSN: 0036-1445. DOI: 10.1137/1037043. URL: http://www.jstor.org/stable/2132823 (cited on page 21).

Hassin, Refael, Shlomi Rubinstein, and Arie Tamir. "Approximation algorithms for maximum dispersion". In: *Operations Research Letters* 21.3 (October 1997), pages 133–137. ISSN: 0167-6377. DOI: 10.1016/S0167-6377(97)00034-5. URL: http://dx.doi.org/10.1016/S0167-6377(97)00034-5 (cited on page 121).

Hernández, Emili, Marc Carreras, and Pere Ridao. "A comparison of homotopic path Planning Algorithms for robotic applications". In: *Robotics and Autonomous Systems* 64 (February 2015), pages 44–58. ISSN: 0921-8890. DOI: 10.1016/j.robot.2014.10.021. URL: http://dx.doi.org/10.1016/j.robot.2014.10.021 (cited on page 11).

Hershberger, John, Matthew Maxel, and Subhash Suri. "Finding the k shortest simple paths: A new algorithm and its implementation". In: *ACM Transactions on Algorithms* 3.4 (November 2007). ISSN: 1549-6325. DOI: 10.1145/1290672.1290682. URL: http://doi.acm.org/10.1145/1290672.1290682 (cited on page 122).

Hershberger, John and Jack Snoeyink. "Computing Minimum Length Paths of a Given Homotopy Class". In: *Computational Geometry* 4 (June 1994), pages 63–97. ISSN: 0925-7721. DOI: 10.1016/0925-7721(94)90010-8. URL: http://dx.doi.org/10.1016/0925-7721(94)90010-8 (cited on page 122).

Hershberger, John and Subhash Suri. "An Optimal Algorithm for Euclidean Shortest Paths in the Plane". In: *SIAM Journal on Computing* 28.6 (1999), pages 2215–2256. ISSN: 0097-5397. DOI: 10.1137/S0097539795289604. URL: http://dx.doi.org/10.1137/S0097539795289604 (cited on page 122).

Ho, Pin-Han, János Tapolcai, and Tibor Cinkler. "Segment shared protection in mesh communications networks with bandwidth guaranteed tunnels". In: *IEEE/ACM Transactions on Networking* 12.6 (December 2004), pages 1105–1118. ISSN: 1063-

6692. DOI: 10.1109/TNET.2004.838592. URL: http://dx.doi.org/10.1109/TNET.2004.838592 (cited on page 123).

Hsu, David, Jean-Claude Latombe, and Hanna Kurniawati. "On the Probabilistic Foundations of Probabilistic Roadmap Planning". In: *International Journal of Robotics Research* 25.7 (July 2006), pages 627–643. ISSN: 0278-3649. DOI: 10.1177/0278364906067174. URL: http://dx.doi.org/10.1177/0278364906067174 (cited on page 10).

Hsu, David, Jean-Claude Latombe, and Rajeev Motwani. "Path Planning in Expansive Configuration Spaces". In: *International Journal of Computational Geometry & Applications* 9.4/5 (August 1999), pages 495–512. ISSN: 0218-1959. DOI: 10.1142/S0218195999000285. URL: http://dx.doi.org/10.1142/S0218195999000285 (cited on page 10).

Hsu, David, Robert Kindel, Jean-Claude Latombe, and Stephen M. Rock. "Randomized Kinodynamic Motion Planning with Moving Obstacles". In: *International Journal of Robotics Research* 21.3 (March 2002), pages 233–256. ISSN: 0278-3649. DOI: 10.1177/027836402320556421. URL: http://dx.doi.org/10.1177/027836402320556421 (cited on page 10).

Jaillet, Léonard and Thierry Siméon. "Path Deformation Roadmaps: Compact Graphs with Useful Cycles for Motion Planning". In: *International Journal of Robotics Research* 27.11-12 (2008), pages 1175–1188. ISSN: 0278-3649. DOI: 10.1177/0278364908098411. URL: http://dx.doi.org/10.1177/0278364908098411 (cited on page 11).

Janson, Lucas, Edward Schmerling, Ashley A. Clark, and Marco Pavone. "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions". In: *International Journal of Robotics Research* 34.7 (June 2015), pages 883–921. ISSN: 0278-3649. DOI: 10.1177/0278364915577958. URL: http://dx.doi.org/10.1177/0278364915577958 (cited on pages 2, 10).

Jean, Frédéric. *Control of Nonholonomic Systems: from Sub-Riemannian Geometry to Motion Planning*. SpringerBriefs in Mathematics. Springer, September 2014. ISBN: 978-3-31908-689-7. DOI: 10.1007/978-3-319-08690-3 (cited on page 12).

Joksch, H.C. "The shortest route problem with constraints". In: *Journal of Mathematical Analysis and Applications* 14.2 (May 1966), pages 191–197. ISSN: 0022-247X. DOI: http://dx.doi.org/10.1016/0022-247X(66)90020-5. URL: http://www.sciencedirect.com/science/article/pii/0022247X66900205 (cited on page 123).

Karaman, Sertac and Emilio Frazzoli. "Sampling-based algorithms for optimal motion planning". In: *International Journal of Robotics Research* 30.7 (June 2011), pages 846–894. ISSN: 0278-3649. DOI: 10.1177/0278364911406761. URL: http://dx.doi.org/10.1177/0278364911406761 (cited on pages 2, 10, 89, 93).

Kavraki, Lydia E., Petr Svestka, Jean-Claude Latombe, and Mark H. Overmars. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces". In: *IEEE Transactions on Robotics and Automation* 12.4 (August 1996), pages 566–580. ISSN: 1042-296X. DOI: 10.1109/70.508439. URL: http://dx.doi.org/10.1109/70.508439 (cited on pages 10, 93).

Kibalczyc, K. and S. Walczak. "Necessary optimality conditions for a problem with costs of rapid variation of control". In: *Journal of the Australian Mathematical Society*. B. Applied Mathematics 26 (01 July 1984), pages 45–55. ISSN: 1446-8735.

DOI: 10.1017/S033427000000432X. URL: http://journals.cambridge.org/article_S033427000000432X (cited on page 26).

Kim, Soonkyum, Subhrajit Bhattacharya, and Vijay Kumar. "Path planning for a tethered mobile robot". In: *IEEE International Conference on Robotics and Automation, ICRA*. May 2014, pages 1132–1139. DOI: 10.1109/ICRA.2014.6906996. URL: http://dx.doi.org/10.1109/ICRA.2014.6906996 (cited on page 11).

Kim, Soonkyum, Koushil Sreenath, Subhrajit Bhattacharya, and Vijay Kumar. "Optimal trajectory generation under homology class constraints". In: *IEEE International Conference on Decision and Control, CDC*. December 2012, pages 3157–3164. DOI: 10.1109/CDC.2012.6425970. URL: http://dx.doi.org/10.1109/CDC.2012.6425970 (cited on page 11).

Kim, Soonkyum, Subhrajit Bhattacharya, Robert Ghrist, and Vijay Kumar. "Topological exploration of unknown and partially known environments". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. November 2013, pages 3851–3858. DOI: 10.1109/IROS.2013.6696907. URL: http://dx.doi.org/10.1109/IROS.2013.6696907 (cited on page 11).

Knepper, Ross A. "On the Fundamental Relationships Among Path Planning Alternatives". PhD thesis. Carnegie Mellon University, June 2011. URL: http://repository.cmu.edu/dissertations/180/ (cited on pages 115, 121, 134).

Knepper, Ross A. and Matthew T. Mason. "Path diversity is only part of the problem". In: *IEEE International Conference on Robotics and Automation, ICRA*. May 2009, pages 3224–3229. DOI: 10.1109/ROBOT.2009.5152696. URL: http://dx.doi.org/10.1109/ROBOT.2009.5152696 (cited on pages 115, 121).

Knepper, Ross A., Siddhartha S. Srinivasa, and Matthew T. Mason. "Toward a deeper understanding of motion alternatives via an equivalence relation on local paths". In: *International Journal of Robotics Research* 31.2 (February 2012), pages 167–186. ISSN: 0278-3649. DOI: 10.1177/0278364911430418. URL: http://dx.doi.org/10.1177/0278364911430418 (cited on pages 115, 121, 124).

Krastanov, Mikhail Ivanov, N. K. Ribarska, and Ts. Y. Tsachev. "A Pontryagin Maximum Principle for Infinite-Dimensional Problems". In: *SIAM Journal Control and Optimization* 49.5 (2011), pages 2155–2182. ISSN: 0363-0129. DOI: 10.1137/100799009. URL: http://dx.doi.org/10.1137/100799009 (cited on page 21).

Krener, Arthur J. "The High Order Maximal Principle and Its Application to Singular Extremals". In: *SIAM Journal Control and Optimization* 15.2 (1977), pages 256–293. ISSN: 0363-0129. DOI: 10.1137/0315023. URL: http://dx.doi.org/10.1137/0315023 (cited on page 21).

Kuby, Michael J. "Programming Models for Facility Dispersion: The p-Dispersion and Maxisum Dispersion Problems". In: *Geographical Analysis* 19.4 (September 1987), pages 315–329. ISSN: 1538-4632. DOI: 10.1111/j.1538-4632.1987.tb00133.x. URL: http://dx.doi.org/10.1111/j.1538-4632.1987.tb00133.x (cited on pages 120, 121).

Kuderer, Markus, Christoph Sprunk, Henrik Kretzschmar, and Wolfram Burgard. "Online generation of homotopically distinct navigation paths". In: *IEEE International Conference on Robotics and Automation, ICRA*. May 2014, pages 6462–6467. DOI: 10.1109/ICRA.2014.6907813. URL: http://dx.doi.org/10.1109/ICRA.2014.6907813 (cited on page 11).

Lafferriere, Gerardo and Héctor J. Sussmann. “A Differential Geometric Approach to Motion Planning”. In: *Nonholonomic Motion Planning*. Edited by Zexiang Li and John F. Canny. Volume 192. The Springer International Series in Engineering and Computer Science. Springer, 1993, pages 235–270. ISBN: 978-1-4613-6392-7. DOI: 10.1007/978-1-4615-3176-0_7 (cited on page 12).

Lafferriere, Gerardo and Héctor J. Sussmann. “Motion planning for controllable systems without drift”. In: *IEEE International Conference on Robotics and Automation, ICRA*. Volume 2. April 1991, pages 1148–1153. ISBN: 0-8186-2163-X. DOI: 10.1109/ROBOT.1991.131763 (cited on page 12).

Laumond, Jean-Paul. *Robot motion planning and control*. Volume 229. Lectures Notes in Control and Information Sciences. Springer, May 1998. ISBN: 978-3-54076-219-5. DOI: 10.1007/BFb0036069 (cited on pages 12, 112).

Laumond, Jean-Paul and Jean-Jacques Risler. “Nonholonomic Systems: Controllability and Complexity”. In: *Theoretical Computer Science* 157.1 (April 1996), pages 101–114. ISSN: 0304-3975. DOI: 10.1016/0304-3975(95)00178-6. URL: http://dx.doi.org/10.1016/0304-3975(95)00178-6 (cited on page 12).

Laumond, Jean-Paul, Paul E. Jacobs, Michel Taïx, and Richard M. Murray. “A motion planner for nonholonomic mobile robots”. In: *IEEE Transactions on Robotics and Automation* 10.5 (October 1994), pages 577–593. ISSN: 1042-296X. DOI: 10.1109/70.326564. URL: http://dx.doi.org/10.1109/70.326564 (cited on page 12).

LaValle, Steven M. *Planning Algorithms*. New York, NY, USA: Cambridge University Press, May 2006. ISBN: 978-0-52186-205-9. DOI: 10.1017/CBO9780511546877 (cited on page 8).

LaValle, Steven M. and James J. Kuffner Jr. “Randomized Kinodynamic Planning”. In: *International Journal of Robotics Research* 20.5 (May 2001), pages 378–400. ISSN: 0278-3649. DOI: 10.1177/02783640122067453. URL: http://dx.doi.org/10.1177/02783640122067453 (cited on pages 10, 16).

LaValle, Steven M. and Prashanth Konkimalla. “Algorithms for Computing Numerical Optimal Feedback Motion Strategies”. In: *International Journal of Robotics Research* 20.9 (September 2001), pages 729–752. ISSN: 0278-3649. DOI: 10.1177/02783640122067633. URL: http://dx.doi.org/10.1177/02783640122067633 (cited on page 24).

Lee, Hyang-Won, Eytan Modiano, and Kayi Lee. “Diverse Routing in Networks With Probabilistic Failures”. In: *IEEE/ACM Transactions on Networking* 18.6 (December 2010), pages 1895–1907. ISSN: 1063-6692. DOI: 10.1109/TNET.2010.2050490. URL: http://dx.doi.org/10.1109/TNET.2010.2050490 (cited on page 124).

Lera, Daniela and Yaroslav D. Sergeyev. “Acceleration of Univariate Global Optimization Algorithms Working with Lipschitz Functions and Lipschitz First Derivatives”. In: *SIAM Journal on Optimization* 23.1 (March 2013), pages 508–529. ISSN: 1052-6234. DOI: 10.1137/110859129. URL: http://dx.doi.org/10.1137/110859129 (cited on page 88).

Li, Chung-Lun, S. Thomas McCormick, and David Simchi-Levi. “Finding disjoint paths with different path-costs: Complexity and algorithms”. In: *Networks* 22.7 (December 1992), pages 653–667. ISSN: 1097-0037. DOI: 10.1002/net.3230220705. URL: http://dx.doi.org/10.1002/net.3230220705 (cited on page 123).

Li, Chung-Lun, S. Thomas McCormick, and David Simchi-Levi. “The complexity of finding two disjoint paths with min-max objective function”. In: *Discrete Applied*

Mathematics 26.1 (January 1990), pages 105–115. ISSN: 0166-218X. DOI: 10.1016/0166-218X(90)90024-7. URL: http://dx.doi.org/10.1016/0166-218X(90)90024-7 (cited on page 123).

Li, Yanbo, Zakary Littlefield, and Kostas E. Bekris. "Asymptotically optimal sampling-based kinodynamic planning". In: *International Journal of Robotics Research* (2016). to appear. ISSN: 0278-3649. DOI: 10.1177/0278364915614386. URL: http://dx.doi.org/10.1177/0278364915614386 (cited on page 93).

Li, Zexiang and John F. Canny. *Nonholonomic Motion Planning*. Volume 192. The Springer International Series in Engineering and Computer Science. Springer, January 1993. ISBN: 978-1-4613-6392-7. DOI: 10.1007/978-1-4615-3176-0 (cited on page 12).

Liberzon, Daniel. *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton University Press, January 2012. ISBN: 978-0-69115-187-8. URL: http://press.princeton.edu/titles/9760.html (cited on page 21).

Lingelbach, Frank. "Path Planning Using Probabilistic Cell Decomposition". In: *IEEE International Conference on Robotics and Automation, ICRA*. April 2004, pages 467–472. DOI: 10.1109/ROBOT.2004.1307193. URL: http://dx.doi.org/10.1109/ROBOT.2004.1307193 (cited on page 10).

Littlefield, Zakary, Yanbo Li, and Kostas E. Bekris. "Efficient sampling-based motion planning with asymptotic near-optimality guarantees for systems with dynamics". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. November 2013, pages 1779–1785. DOI: 10.1109/IROS.2013.6696590. URL: http://dx.doi.org/10.1109/IROS.2013.6696590 (cited on page 93).

Loxton, Ryan C., Qun Lin, and Kok Lay Teo. “Minimizing control variation in nonlinear optimal control”. In: *Automatica* 49.9 (September 2013), pages 2652–2664. ISSN: 0005-1098. DOI: 10.1016/j.automatica.2013.05.027. URL: http://dx.doi.org/10.1016/j.automatica.2013.05.027 (cited on page 26).

Lyu, Yu-Han and Devin J. Balkcom. “Optimal Trajectories for Kinematic Planar Rigid Bodies with Switching Costs”. In: *International Journal of Robotics Research* (2016). to appear. ISSN: 0278-3649. DOI: 10.1177/0278364915594243. URL: http://dx.doi.org/10.1177/0278364915594243 (cited on pages 3, 24, 25).

Lyu, Yu-Han and Devin J. Balkcom. “Optimal Trajectories for Planar Rigid Bodies with Switching Costs”. In: *Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics*. Volume 107. Springer Tracts in Advanced Robotics. 2014, pages 371–387. ISBN: 978-3-319-16594-3. DOI: 10.1007/978-3-319-16595-0_22. URL: http://dx.doi.org/10.1007/978-3-319-16595-0_22 (cited on pages 3, 24).

Lyu, Yu-Han, Yining Chen, and Devin J. Balkcom. “k-survivability: Diversity and survival of expendable robots”. In: *IEEE Robotics and Automation Letters (RA-L)* (2016). to appear (cited on page 6).

Lyu, Yu-Han, Yining Chen, and Devin J. Balkcom. “k-survivability: Diversity and survival of expendable robots”. In: *IEEE International Conference on Robotics and Automation, ICRA*. to appear. 2016 (cited on page 6).

Lyu, Yu-Han, Andrei A. Furtuna, Weifu Wang, and Devin J. Balkcom. “The bench mover’s problem: Minimum-time trajectories, with cost for switching between controls”. In: *IEEE International Conference on Robotics and Automation, ICRA*.

June 2014, pages 106–112. DOI: 10.1109/ICRA.2014.6906596. URL: http://dx.doi.org/10.1109/ICRA.2014.6906596 (cited on pages 3, 23, 86).

Marble, James D. and Kostas E. Bekris. "Asymptotically Near-Optimal Planning With Probabilistic Roadmap Spanners". In: *IEEE Transactions on Robotics* 29.2 (April 2013), pages 432–444. ISSN: 1552-3098. DOI: 10.1109/TRO.2012.2234312. URL: http://dx.doi.org/10.1109/TRO.2012.2234312 (cited on page 93).

Martí, Rafael, Micael Gallego, Abraham Duarte, and Eduardo G. Pardo. "Heuristics and metaheuristics for the maximum diversity problem". In: *Journal of Heuristics* 19.4 (August 2013), pages 591–615. ISSN: 1381-1231. DOI: 10.1007/s10732-011-9172-4. URL: http://dx.doi.org/10.1007/s10732-011-9172-4 (cited on pages 121, 135).

Mason, Matthew T. *Mechanics of Robotic Manipulation*. Intelligent Robotics and Autonomous Agents series. The MIT Press, June 2001. ISBN: 978-0-26213-396-8. URL: http://mitpress.mit.edu/books/mechanics-robotic-manipulation (cited on pages 86, 97).

Matula, Joanna. "On an extremum problem". In: *Journal of the Australian Mathematical Society*. B. Applied Mathematics 28 (03 January 1987), pages 376–392. ISSN: 1446-8735. DOI: 10.1017/S0334270000005464. URL: http://journals.cambridge.org/article_S0334270000005464 (cited on page 26).

Maurer, Helmut. "On Optimal Control Problems with Bounded State Variables and Control Appearing Linearly". In: *SIAM Journal on Control and Optimization* 15.3 (1977), pages 345–362. ISSN: 0363-0129. DOI: 10.1137/0315023. URL: http://dx.doi.org/10.1137/0315023 (cited on page 88).

Maurer, Helmut and Nikolai P. Osmolovskii. "Second-order conditions for optimal control problems with mixed control-state constraints and control appearing linearly". In: *IEEE International Conference on Decision and Control, CDC*. December 2013, pages 514–519. DOI: 10.1109/CDC.2013.6759933. URL: http://dx.doi.org/10.1109/CDC.2013.6759933 (cited on page 88).

McCarthy, Zoe, Timothy Bretl, and Seth Hutchinson. "Proving path non-existence using sampling and alpha shapes". In: *IEEE International Conference on Robotics and Automation, ICRA*. May 2012, pages 2563–2569. DOI: 10.1109/ICRA.2012.6225300. URL: http://dx.doi.org/10.1109/ICRA.2012.6225300 (cited on pages 11, 105).

Miller, Boris, Karen Stepanyan, Alexander Miller, and Michael Andreev. "3D path planning in a threat environment". In: *IEEE Conference on Decision and Control and European Control Conference, CDC-ECC*. December 2011, pages 6864–6869. ISBN: 978-1-61284-800-6. DOI: 10.1109/CDC.2011.6160385. URL: http://dx.doi.org/10.1109/CDC.2011.6160385 (cited on pages 3, 122, 141).

Milyutin, A. A. and N. P. Osmolovskii. *Calculus of Variations and Optimal Control*. Volume 180. Translations of Mathematical Monographs. American Mathematical Society, August 1998. ISBN: 978-0-82180-753-8. URL: http://www.ams.org/bookstore-getitem/item=MMONO-180 (cited on page 21).

Montgomery, Richard. *A Tour of Subriemannian Geometries, Their Geodesics and Applications*. Volume 91. Mathematical Surveys and Monographs. American Mathematical Society, January 2002. ISBN: 978-0-82184-165-5. DOI: 10.1090/surv/091 (cited on page 12).

Moon, I. Douglas and Sohail S. Chaudhry. "An Analysis of Network Location Problems with Distance Constraints". In: *Management Science* 30.3 (March 1984), pages 290–307. ISSN: 0025-1909. DOI: 10.1287/mnsc.30.3.290. URL: http://dx.doi.org/10.1287/mnsc.30.3.290 (cited on page 120).

Moutarlier, Philippe, Brian Mirtich, and John F. Canny. "Shortest Paths for a Car-like Robot to Manifolds in Configuration Space". In: *International Journal of Robotics Research* 15.1 (February 1996), pages 36–60. ISSN: 0278-3649. DOI: 10.1177/027836499601500103. URL: http://dx.doi.org/10.1177/027836499601500103 (cited on page 23).

Murphy, Robin R. *Disaster Robotics*. Intelligent Robotics and Autonomous Agents series. The MIT Press, February 2014. ISBN: 978-0-26202-735-9. URL: https://mitpress.mit.edu/books/disaster-robotics (cited on page 140).

Murray, Richard M. and S. Shankar Sastry. "Nonholonomic motion planning: steering using sinusoids". In: *IEEE Transactions on Automatic Control* 38.5 (May 1993), pages 700–716. ISSN: 0018-9286. DOI: 10.1109/9.277235 (cited on page 12).

Nieuwenhuisen, Dennis and Mark H. Overmars. "Useful Cycles in Probabilistic Roadmap Graphs". In: *IEEE International Conference on Robotics and Automation, ICRA*. April 2004, pages 446–452. DOI: 10.1109/ROBOT.2004.1307190. URL: http://dx.doi.org/10.1109/ROBOT.2004.1307190 (cited on page 11).

Noussair, E.S. "On the existence of piecewise continuous optimal controls". In: *Journal of the Australian Mathematical Society*. B. Applied Mathematics 20 (01 June 1977), pages 31–37. ISSN: 1446-8735. DOI: 10.1017/S0334270000001429. URL: http://journals.cambridge.org/article_S0334270000001429 (cited on page 26).

Ny, Jerome Le, Eric Feron, and Emilio Frazzoli. “On the Dubins Traveling Salesman Problem”. In: *IEEE Transactions on Automatic Control* 57.1 (January 2012), pages 265–270. ISSN: 0018-9286. DOI: 10.1109/TAC.2011.2166311. URL: http://dx.doi.org/10.1109/TAC.2011.2166311 (cited on page 22).

Ó'Dúnlaing, Colm. “Motion Planning with Inertial Constraints”. In: *Algorithmica* 2 (November 1987), pages 431–475. ISSN: 0178-4617. DOI: 10.1007/BF01840370. URL: http://dx.doi.org/10.1007/BF01840370 (cited on page 7).

Omran, Masoud Taghinezhad, Jörg-Rüdiger Sack, and Hamid Zarrabi-Zadeh. “Finding paths with minimum shared edges”. In: *Journal of Combinatorial Optimization* 26.4 (November 2013), pages 709–722. ISSN: 1382-6905. DOI: 10.1007/s10878-012-9462-2. URL: http://dx.doi.org/10.1007/s10878-012-9462-2 (cited on page 123).

Osmolovskii, Nikolai P. and Helmut Maurer. *Applications to Regular and Bang-Bang Control - Second-Order Necessary and Sufficient Optimality Conditions in Calculus of Variations and Optimal Control*. Volume 24. Advances in design and control. SIAM, 2012. ISBN: 978-1-611-97235-1. URL: http://www.ec-securehost.com/SIAM/DC24.html (cited on page 21).

Otte, Michael and Emilio Frazzoli. “RRTX: Asymptotically optimal single-query sampling-based motion planning with quick replanning”. In: *International Journal of Robotics Research* (2016). ISSN: 0278-3649. DOI: 10.1177/0278364915594679. URL: http://dx.doi.org/10.1177/0278364915594679 (cited on pages 2, 10).

Pintér, János D. *Global Optimization in Action: Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications*. Volume 6. Nonconvex Optimization and Its Applications. Springer, 1996. ISBN: 978-0-7923-3757-7. DOI: 10.1007/978-1-4757-2502-5 (cited on page 59).

Piyavskii, S. A. "An algorithm for finding the absolute minimum of a function". In: *USSR Computational Mathematics and Mathematical Physics* 12 (4 1967), pages 13–24 (cited on page 59).

Pokorny, Florian T., Majd Hawasly, and Subramanian Ramamoorthy. "Topological trajectory classification with filtrations of simplicial complexes and persistent homology". In: *International Journal of Robotics Research* 35.1-3 (2016), pages 204–223. ISSN: 0278-3649. DOI: 10.1177/0278364915586713. URL: http://dx.doi.org/10.1177/0278364915586713 (cited on page 11).

Pontryagin, Lev Semyonovich, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *Mathematical Theory of Optimal Processes*. John Wiley & Sons Inc, January 1962. ISBN: 978-0-47069-381-0 (cited on pages 17, 21, 23, 26, 28).

Porta, Josep M., Juan Cortés, Lluís Ros, and Federico Thomas. "A space decomposition method for path planning of loop linkages". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. October 2007, pages 1882–1888. ISBN: 978-1-4244-0912-9. DOI: 10.1109/IROS.2007.4399140. URL: http://dx.doi.org/10.1109/IROS.2007.4399140 (cited on page 9).

Quispe, Ana Huaman, Tobias Kunz, and Mike Stilman. "Generation of diverse paths in 3D environments". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. November 2013, pages 5994–5999. DOI: 10.1109/IROS.2013.6697226. URL: http://dx.doi.org/10.1109/IROS.2013.6697226 (cited on pages 115, 121).

Ravi, S. S., Daniel J. Rosenkrantz, and Giri Kumar Tayi. “Heuristic and Special Case Algorithms for Dispersion Problems”. In: *Operations Research* 42.2 (March 1994), pages 299–310. ISSN: 0030-364X. URL: http://www.jstor.org/stable/171673 (cited on pages 120, 121).

Reeds, James Alexander and Lawrence A. Shepp. “Optimal paths for a car that goes both forwards and backwards”. In: *Pacific Journal of Mathematics* 145.2 (1990), pages 367–393. DOI: 10.2140/pjm.1990.145.367. URL: http://dx.doi.org/10.2140/pjm.1990.145.367 (cited on pages 2, 4, 15, 22, 91).

Reif, John H. and Stephen R. Tate. “Approximate kinodynamic planning using L2-norm dynamic bounds”. In: *Computers & Mathematics with Applications* 27.5 (March 1994), pages 29–44. ISSN: 0898-1221. DOI: 10.1137/S0097539798331975. URL: http://dx.doi.org/10.1137/S0097539798331975 (cited on page 92).

Reif, John H. and Hongyan Wang. “Nonuniform Discretization for Kinodynamic Motion Planning and its Applications”. In: *SIAM Journal on Computing* 30.1 (2000), pages 161–190. ISSN: 0097-5397. DOI: 10.1137/S0097539798331975. URL: http://dx.doi.org/10.1137/S0097539798331975 (cited on page 92).

Reif, John H. and Hongyan Wang. “The complexity of the two dimensional curvature-constrained shortest-path problem”. In: *Workshop on the Algorithmic Foundations of Robotics, WAFR*. Edited by Pankaj K. Agarwal, Lydia E. Kavraki, and Matt Mason. Natick, MA, USA: A.K. Peters, Ltd., March 1998, pages 49–57. ISBN: 1-56881-081-4. URL: http://www.cs.duke.edu/~reif/paper/hongyan/curve/curve.pdf (cited on page 22).

Reister, David B. and François G. Pin. “Time-Optimal Trajectories for Mobile Robots With Two Independently Driven Wheels”. In: *International Journal of Robotics*

Research 13.1 (February 1994), pages 38–54. ISSN: 0278-3649. DOI: 10.1177/027836499401300103. URL: http://dx.doi.org/10.1177/027836499401300103 (cited on page 23).

Renaud, Marc and Jean-Yves Fourquet. “Minimum time motion of a mobile robot with two independent, acceleration-driven wheels”. In: *IEEE International Conference on Robotics and Automation, ICRA*. Volume 3. April 1997, pages 2608–2613. ISBN: 0-7803-3612-7. DOI: 10.1109/ROBOT.1997.619354. URL: http://dx.doi.org/10.1109/ROBOT.1997.619354 (cited on page 23).

Roditty, Liam. “On the k Shortest Simple Paths Problem in Weighted Directed Graphs”. In: *SIAM Journal on Computing* 39.6 (2010), pages 2363–2376. ISSN: 0097-5397. DOI: 10.1137/080730950. URL: http://dx.doi.org/10.1137/080730950 (cited on page 122).

Rohrer, Justin P., Abdul Jabbar, and James P. G. Sterbenz. “Path diversification for future internet end-to-end resilience and survivability”. In: *Telecommunication Systems* 56.1 (May 2014), pages 49–67. ISSN: 1018-4864. DOI: 10.1007/s11235-013-9818-7. URL: http://dx.doi.org/10.1007/s11235-013-9818-7 (cited on page 124).

Rosell, Jan and Pedro Iñiguez. “Path planning using Harmonic Functions and Probabilistic Cell Decomposition”. In: *IEEE International Conference on Robotics and Automation, ICRA*. April 2005, pages 1803–1808. ISBN: 0-7803-8914-X. DOI: 10.1109/ROBOT.2005.1570375. URL: http://dx.doi.org/10.1109/ROBOT.2005.1570375 (cited on page 10).

Ross, I. Michael and Mark Karpenko. “A review of pseudospectral optimal control: From theory to flight”. In: *Annual Reviews in Control* 36.2 (December 2012), pages 182–197. ISSN: 1367-5788. DOI: 10.1016/j.arcontrol.2012.09.002. URL: http://dx.doi.org/10.1016/j.arcontrol.2012.09.002 (cited on page 25).

Rubenstein, Michael, Christian Ahler, Nick Hoff, Adrian Cabrera, and Radhika Nagpal. "Kilobot: A low cost robot with scalable operations designed for collective behaviors". In: *Robotics and Autonomous Systems* 62.7 (July 2014), pages 966–975. ISSN: 0921-8890. DOI: 10.1016/j.robot.2013.08.006. URL: http://dx.doi.org/10.1016/j.robot.2013.08.006 (cited on page 140).

Salzman, Oren and Dan Halperin. "Asymptotically near-optimal RRT for fast, high-quality, motion planning". In: *IEEE International Conference on Robotics and Automation, ICRA*. May 2014, pages 4680–4685. DOI: 10.1109/ICRA.2014.6907543. URL: http://dx.doi.org/10.1109/ICRA.2014.6907543 (cited on pages 2, 10).

Salzman, Oren, Doron Shaharabani, Pankaj K. Agarwal, and Dan Halperin. "Sparsification of motion-planning roadmaps by edge contraction". In: *International Journal of Robotics Research* 33.14 (November 2014), pages 1711–1725. ISSN: 0278-3649. DOI: 10.1177/0278364914556517. URL: http://dx.doi.org/10.1177/0278364914556517 (cited on page 93).

Schättler, Heinz and Urszula Ledzewicz. *Geometric Optimal Control: Theory, Methods and Examples*. Volume 38. Interdisciplinary Applied Mathematics. Springer, July 2012. ISBN: 978-1-4614-3833-5. DOI: 10.1007/978-1-4614-3834-2 (cited on page 20).

Schmitzberger, Erwin, Jean-Louis Bouchet, Michel Dufaut, Didier Wolf, and René Husson. "Capture of homotopy classes with probabilistic road map". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. September

2002, pages 2317–2322. ISBN: 0-7803-7398-7. DOI: 10.1109/IRDS.2002.1041613. URL: http://dx.doi.org/10.1109/IRDS.2002.1041613 (cited on page 11).

Schwartz, Jacob T and Micha Sharir. "On the piano movers'Ăİ problem. II. General techniques for computing topological properties of real algebraic manifolds". In: *Advances in Applied Mathematics* 4.3 (September 1983), pages 298 –351. ISSN: 0196-8858. DOI: http://dx.doi.org/10.1016/0196-8858(83)90014-3. URL: http://www.sciencedirect.com/science/article/pii/0196885883900143 (cited on page 7).

Sekhavat, Sepanta, Petr Svestka, Jean-Paul Laumond, and Mark H. Overmars. "Multilevel Path Planning for Nonholonomic Robots Using Semiholonomic Subsystems". In: *International Journal of Robotics Research* 17.8 (August 1998), pages 840–857. ISSN: 0278-3649. DOI: 10.1177/027836499801700803. URL: http://dx.doi.org/10.1177/027836499801700803 (cited on page 12).

Shkel, Andrei M. and Vladimir J. Lumelsky. "Classification of the Dubins set". In: *Robotics and Autonomous Systems* 34.4 (March 2001), pages 179–202. ISSN: 0921-8890. DOI: 10.1016/S0921-8890(00)00127-5. URL: http://dx.doi.org/10.1016/S0921-8890(00)00127-5 (cited on page 22).

Shvalb, Nir, Moshe Shoham, Guanfeng Liu, and Jeffrey C. Trinkle. "Motion Planning for a Class of Planar Closed-chain Manipulators". In: *International Journal of Robotics Research* 26.5 (May 2007), pages 457–473. ISSN: 0278-3649. DOI: 10.1177/0278364907078094. URL: http://dx.doi.org/10.1177/0278364907078094 (cited on page 7).

Siméon, Thierry, Jean-Paul Laumond, and Carole Nissoux. "Visibility-based probabilistic roadmaps for motion planning". In: *Advanced Robotics* 14.6 (2000), pages 477–

493. DOI: 10.1163/156855300741960. URL: http://dx.doi.org/10.1163/156855300741960 (cited on page 10).

Sommer, Christian. "Shortest-path queries in static networks". In: *ACM Computing Surveys* 46.4 (April 2014), 45:1–45:31. ISSN: 0360-0300. DOI: 10.1145/2530531. URL: http://doi.acm.org/10.1145/2530531 (cited on page 93).

Souères, Philippe. "Minimum-Length Trajectories for a Car: An Example of the Use of Boltianskii's Sufficient Conditions for Optimality". In: *IEEE Transactions on Automatic Control* 52.2 (February 2007), pages 323–327. ISSN: 0018-9286. DOI: 10.1109/TAC.2006.890319. URL: http://dx.doi.org/10.1109/TAC.2006.890319 (cited on page 23).

Souères, Philippe and Jean-Daniel Boissonnat. "Optimal Trajectories for Nonholonomic Mobile Robots". In: *Robot Motion Planning and Control*. Edited by Jean-Paul Laumond. Volume 229. Lecture Notes in Control and Information Sciences. Springer, May 1998, pages 93–170. ISBN: 978-3-540-76219-5. DOI: 10.1007/BFb0036072. URL: http://dx.doi.org/10.1007/BFb0036072 (cited on page 23).

Souères, Philippe and Jean-Paul Laumond. "Shortest paths synthesis for a car-like robot". In: *IEEE Transactions on Automatic Control* 41.5 (May 1996), pages 672–688. ISSN: 0018-9286. DOI: 10.1109/9.277235 (cited on page 23).

Stefanakos, Stamatis. "Reliable routings in networks with generalized link failure events". In: *IEEE/ACM Transactions on Networking* 16.6 (December 2008), pages 1331–1339. ISSN: 1063-6692. DOI: 10.1145/1514016.1514024. URL: http://doi.acm.org/10.1145/1514016.1514024 (cited on page 123).

Stewart, David E. "A numerical algorithm for optimal control problems with switching costs". In: *Journal of the Australian Mathematical Society*. B. Applied Mathematics 34 (02 October 1992), pages 212–228. ISSN: 1446-8735. DOI: 10.1017/S0334270000008730. URL: http://journals.cambridge.org/article_S0334270000008730 (cited on pages 16, 26).

Sussmann, Héctor J. “Set-valued differentials and the hybrid maximum principle”. In: *IEEE International Conference on Decision and Control, CDC*. Volume 1. December 2000, pages 558–563. DOI: 10.1109/CDC.2000.912823 (cited on pages 21, 26).

Sussmann, Héctor J. “The Markov-Dubins problem with angular acceleration control”. In: *IEEE International Conference on Decision and Control, CDC*. Volume 3. December 1997, pages 2639–2643. DOI: 10.1109/CDC.1997.657778 (cited on pages 15, 22).

Sussmann, Héctor J. and Guoqing Tang. *Shortest Paths for the Reeds-Shepp Car: A Worked Out Example of the Use Of Geometric Techniques In Nonlinear Optimal Control.* SYCON 91-10. New Brunswick, NJ 08903: Department of Mathematics, Rutgers University, 1991. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.8921 (cited on page 23).

Suurballe, J. W. “Disjoint paths in a network”. In: *Networks* 4.2 (1974), pages 125–145. ISSN: 1097-0037. DOI: 10.1002/net.3230040204. URL: http://dx.doi.org/10.1002/net.3230040204 (cited on page 123).

Suurballe, J. W. and Robert Endre Tarjan. “A quick method for finding shortest pairs of disjoint paths”. In: *Networks* 14.2 (1984), pages 325–336. ISSN: 1097-0037. DOI: 10.1002/net.3230140209. URL: http://dx.doi.org/10.1002/net.3230140209 (cited on page 123).

Svestka, Petr and Mark H. Overmars. “Motion Planning for Carlike Robots Using a Probabilistic Learning Approach”. In: *International Journal of Robotics Research* 16.2 (April 1997), pages 119–143. ISSN: 0278-3649. DOI: 10.1177/027836499701600201. URL: http://dx.doi.org/10.1177/027836499701600201 (cited on page 10).

Takei, Ryo and Richard Tsai. “Optimal Trajectories of Curvature Constrained Motion in the Hamilton-Jacobi Formulation”. In: *Journal of Scientific Computing* 54.2-3 (February 2013), pages 622–644. ISSN: 0885-7474. DOI: 10.1007/s10915-012-9671-y. URL: http://dx.doi.org/10.1007/s10915-012-9671-y (cited on page 24).

Tamir, Arie. “Obnoxious Facility Location on Graphs”. In: *SIAM Journal on Discrete Mathematics* 4.4 (1991), pages 550–567. ISSN: 0895-4801. DOI: 10.1137/0404048. URL: http://dx.doi.org/10.1137/0404048 (cited on page 120).

Teo, Kok Lay and Les S. Jennings. “Optimal control with a cost on changing control”. In: *Journal of Optimization Theory and Applications* 68.2 (February 1991), pages 335–357. ISSN: 0022-3239. DOI: 10.1007/BF00941572. URL: http://dx.doi.org/10.1007/BF00941572 (cited on page 26).

Trinkle, Jeffrey C. and R. James Milgram. “Complete Path Planning for Closed Kinematic Chains with Spherical Joints”. In: *International Journal of Robotics Research* 21.9 (September 2002), pages 773–790. ISSN: 0278-3649. DOI: 10.1177/0278364902021009119. URL: http://dx.doi.org/10.1177/0278364902021009119 (cited on page 7).

Varadhan, Gokul, Shankar Krishnan, T. V. N. Sriram, and Dinesh Manocha. “A Simple Algorithm for Complete Motion Planning of Translating Polyhedral Robots”.

In: *International Journal of Robotics Research* 25.11 (November 2006), pages 1049–1070. ISSN: 0278-3649. DOI: 10.1177/0278364906071199. URL: http://dx.doi.org/10.1177/0278364906071199 (cited on page 7).

Vendittelli, Marilena, Jean-Paul Laumond, and Carole Nissoux. "Obstacle distance for car-like robots". In: *IEEE Transactions on Robotics and Automation* 15.4 (August 1999), pages 678–691. ISSN: 1042-296X. DOI: 10.1109/70.781973. URL: http://dx.doi.org/10.1109/70.781973 (cited on page 23).

Verdière, Éric Colin de and Jeff Erickson. "Tightening Nonsimple Paths and Cycles on Surfaces". In: *SIAM Journal on Computing* 39.8 (2010), pages 3784–3813. ISSN: 0097-5397. DOI: 10.1137/090761653. URL: http://dx.doi.org/10.1137/090761653 (cited on page 122).

Voss, Caleb, Mark Moll, and Lydia E. Kavraki. "A heuristic approach to finding diverse short paths". In: *IEEE International Conference on Robotics and Automation, ICRA*. May 2015, pages 4173–4179. DOI: 10.1109/ICRA.2015.7139774. URL: http://dx.doi.org/10.1109/ICRA.2015.7139774 (cited on pages 115, 121, 134, 135).

Wang, Cong, Yi-Jen Chiang, and Chee-Keng Yap. "On soft predicates in subdivision motion planning". In: *Computational Geometry* 48.8 (September 2015), pages 589–605. ISSN: 0925-7721. DOI: 10.1016/j.comgeo.2015.04.002. URL: http://dx.doi.org/10.1016/j.comgeo.2015.04.002 (cited on page 90).

Wang, Huifang, Yangzhou Chen, and Philippe Souères. "A Geometric Algorithm to Compute Time-Optimal Trajectories for a Bidirectional Steered Robot". In: *IEEE Transactions on Robotics* 25.2 (April 2009), pages 399–413. ISSN: 1552-3098. DOI:

10.1109/TRO.2009.2015610. URL: http://dx.doi.org/10.1109/TRO.2009.2015610 (cited on page 23).

Wang, Weifu and Devin J. Balkcom. “Analytical time-optimal trajectories for an omni-directional vehicle”. In: *IEEE International Conference on Robotics and Automation, ICRA*. May 2012, pages 4519–4524. DOI: 10.1109/ICRA.2012.6224602. URL: http://dx.doi.org/10.1109/ICRA.2012.6224602 (cited on pages 23, 86, 87).

Wang, Weifu and Devin J. Balkcom. “Sampling Extremal Trajectories for Planar Rigid Bodies”. In: *Workshop on the Algorithmic Foundations of Robotics, WAFR*. Volume 86. Springer Tracts in Advanced Robotics. June 2012, pages 331–347. ISBN: 978-3-642-36278-1. DOI: 10.1007/978-3-642-36279-8_20. URL: http://dx.doi.org/10.1007/978-3-642-36279-8_20 (cited on page 23).

Wang, Weifu, Devin J. Balkcom, and Amit Chakrabarti. “A fast online spanner for roadmap construction”. In: *International Journal of Robotics Research* 34.11 (September 2015), pages 1418–1432. ISSN: 0278-3649. DOI: 10.1177/0278364915576491. URL: http://dx.doi.org/10.1177/0278364915576491 (cited on page 93).

White, Douglas J. “The maximal dispersion problem and the "First point outside the neighbourhood" heuristic”. In: *Computers & Operations Research* 18.1 (1991), pages 43–50. ISSN: 0305-0548. DOI: 10.1016/0305-0548(91)90040-X. URL: http://dx.doi.org/10.1016/0305-0548(91)90040-X (cited on page 120).

Whitehead, J. H. C. “Convex Regions in the Geometry of Paths”. In: *The Quarterly Journal of Mathematics* os-4.1 (1932), pages 33–42. ISSN: 0033-5606. DOI: 10.1093/qmath/os-3.1.33. URL: http://qjmath.oxfordjournals.org/content/os-3/1/33 (cited on page 94).

Xu, Dahai, Yang Chen, Yizhi Xiong, Chunming Qiao, and Xin He. "On the complexity of and algorithms for finding the shortest path with a disjoint counterpart". In: *IEEE/ACM Transactions on Networking* 14.1 (February 2006), pages 147–158. ISSN: 1063-6692. DOI: 10.1145/1133553.1133565. URL: http://doi.acm.org/10.1145/1133553.1133565 (cited on pages 116, 123).

Yeh, Hsin-Yi, Shawna L. Thomas, David Eppstein, and Nancy M. Amato. "UOBPRM: A uniformly distributed obstacle-based PRM". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. October 2012, pages 2655–2662. ISBN: 978-1-4673-1737-5. DOI: 10.1109/IROS.2012.6385875. URL: http://dx.doi.org/10.1109/IROS.2012.6385875 (cited on page 92).

Yen, Jin Y. "Finding the K Shortest Loopless Paths in a Network". In: *Management Science* 17.11 (July 1971), pages 712–716. ISSN: 0025-1909. DOI: 10.1287/mnsc.17.11.712. URL: http://dx.doi.org/10.1287/mnsc.17.11.712 (cited on page 122).

Yershov, Dmitry S. and Emilio Frazzoli. "Asymptotically optimal feedback planning using a numerical Hamilton-Jacobi- Bellman solver and an adaptive mesh refinement". In: *International Journal of Robotics Research* (2016). ISSN: 0278-3649. DOI: 10.1177/0278364915602958. URL: http://dx.doi.org/10.1177/0278364915602958 (cited on page 24).

Yershov, Dmitry S. and Steven M. LaValle. "Sufficient Conditions for the Existence of Resolution Complete Planning Algorithms". In: *Workshop on the Algorithmic Foundations of Robotics, WAFR*. Volume 68. December 2010, pages 303–320. ISBN: 978-3-642-17451-3. DOI: 10.1007/978-3-642-17452-0_18. URL: http://dx.doi.org/10.1007/978-3-642-17452-0_18 (cited on page 9).

Yu, Changjun, Bin Li, Ryan C. Loxton, and Kok Lay Teo. "Optimal discrete-valued control computation". In: *Journal of Global Optimization,* 56.2 (June 2013), pages 503–518. ISSN: 0925-5001. DOI: 10.1007/s10898-012-9858-7. URL: http://dx.doi.org/10.1007/s10898-012-9858-7 (cited on page 26).

Yuan, Shengli, Saket Varma, and Jason P. Jue. "Minimum-color path problems for reliability in mesh networks". In: *IEEE International Conference on Computer and Communications, INFOCOM*. March 2005, pages 2658–2669. DOI: 10.1109/INFCOM.2005.1498549. URL: http://dx.doi.org/10.1109/INFCOM.2005.1498549 (cited on page 126).

Yuan, Shengli and Bin Wang. "Highly Available Path Routing in Mesh Networks Under Multiple Link Failures". In: *IEEE Transactions on Reliability* 60.4 (December 2011), pages 823–832. ISSN: 0018-9529. DOI: 10.1109/TR.2011.2171097. URL: http://dx.doi.org/10.1109/TR.2011.2171097 (cited on page 124).

Zabarankin, Michael, Stan Uryasev, and Robert Murphey. "Aircraft routing under the risk of detection". In: *Naval Research Logistics (NRL)* 53.8 (December 2006), pages 728–747. ISSN: 1520-6750. DOI: 10.1002/nav.20165. URL: http://dx.doi.org/10.1002/nav.20165 (cited on pages 3, 122, 141).

Zefran, Milos and Vijay Kumar. "A variational calculus framework for motion planning". In: *International Conference on Advanced Robotics*. July 1997, pages 415–420. ISBN: 0-7803-4160-0. DOI: 10.1109/ICAR.1997.620215 (cited on page 12).

Zelikin, Michail I. and Vladimir F. Borisov. *Theory of Chattering Control: with applications to Astronautics, Robotics, Economics, and Engineering*. Systems & Control: Foundations & Applications. Birkhäuser Basel, April 1994. ISBN: 978-1-4612-7634-0. DOI: 10.1007/978-1-4612-2702-1 (cited on page 13).

Zhang, Huili, Yin-Feng Xu, and Xingang Wen. “Optimal shortest path set problem in undirected graphs”. In: *Journal of Combinatorial Optimization* 29.3 (April 2015), pages 511–530. ISSN: 1382-6905. DOI: 10.1007/s10878-014-9766-5. URL: http://dx.doi.org/10.1007/s10878-014-9766-5 (cited on page 123).

Zhang, Liangjun, Young J. Kim, and Dinesh Manocha. “A hybrid approach for complete motion planning”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. October 2007, pages 7–14. ISBN: 978-1-4244-0912-9. DOI: 10.1109/IROS.2007.4399064. URL: http://dx.doi.org/10.1109/IROS.2007.4399064 (cited on page 9).

Zheng, Si-Qing, Jianping Wang, Bing Yang, and Mei Yang. “Minimum-Cost Multiple Paths Subject to Minimum Link and Node Sharing in a Network”. In: *IEEE/ACM Transactions on Networking* 18.5 (October 2010), pages 1436–1449. ISSN: 1063-6692. DOI: 10.1109/TNET.2010.2044514. URL: http://dx.doi.org/10.1109/TNET.2010.2044514 (cited on page 123).

Zhu, David and Jean-Claude Latombe. “New heuristic algorithms for efficient hierarchical path planning”. In: *IEEE Transactions on Robotics and Automation* 7.1 (February 1991), pages 9–20. ISSN: 1042-296X. DOI: 10.1109/70.68066. URL: http://dx.doi.org/10.1109/70.68066 (cited on page 9).

Zotkiewicz, Mateusz, Walid Ben-Ameur, and Michal Pióro. “Finding Failure-Disjoint Paths for Path Diversity Protection in Communication Networks”. In: *IEEE Communications Letters* 14.8 (August 2010), pages 776–778. ISSN: 1089-7798. DOI: 10.1109/LCOMM.2010.08.100653. URL: http://dx.doi.org/10.1109/LCOMM.2010.08.100653 (cited on page 123).