

Tethered Motion Planning for a Rappelling Robot

Thesis by
Melissa M. Tanner

In Partial Fulfillment of the Requirements for the
degree of
Doctor of Philosophy

The logo for the California Institute of Technology (Caltech), featuring the word "Caltech" in a bold, orange, sans-serif font.

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2020
Defended June 12, 2019

© 2020

Melissa M. Tanner

ORCID: 0000-0003-4610-1379

Some rights reserved. This thesis is distributed under a
Attribution-NonCommercial 4.0 International License

For my family.
And for all the little and not-so-little girls
who also want to play with airplanes and rockets and robots.

ACKNOWLEDGEMENTS

This research would not have been possible without the financial support of the NASA Jet Propulsion Laboratory (JPL) and the Keck Institute for Space Studies (KISS). Working on a NASA project and meeting Buzz Aldrin were childhood dreams come true.

I am forever indebted to Professor Joel Burdick; he has supported my career at every step, from mentoring my summer internship as a high school sophomore to coaching me through writing this thesis. His patience and compassion have taught me how to navigate tough projects and tough times in life. If I can be half the mentor he is, I will consider my career a success.

I am deeply grateful to my JPL supervisor and committee member Dr. Issa Nesnas for always making time in his busy schedule to puzzle through a thorny research problem with me. He showed me by example how to manage a project and a team. I would also like to express my gratitude to my dissertation committee members, Professor Richard Murray and Professor Aaron Ames, for their time and insight.

I would like to acknowledge the contributions of the past and present members of the Axel team at JPL, many of whom I've worked closely with over the course of my research. I'd also like to thank my colleagues in the robotics lab groups for their encouragement and timely distractions. I owe a great debt to Pablo Abad-Manterola, who taught me everything I know about Axel and left very big shoes to fill. More than anything, though, I want to express my heartfelt gratitude to Jeffrey Edlund, my mentor, officemate, colleague, and friend. I could never have completed this project without his theoretical insights, coding help, spare computer parts, and life advice. His wife, Jian Yuan, contributed many cooked meals to the completion of this thesis.

So many people have touched my life and put me on the path to where I am now. Professor AnnMarie Polsenberg Thomas mentored my first long-ago summer internship when she was still a graduate student in Joel's lab. Professors Daniela Rus and Seth Teller (the latter unfortunately deceased) ignited my passion for AI and encouraged me to return to robotics after MIT. I would also like to express my appreciation for the sweet, supportive folks at the Caltech SURF office, as well as all the Caltech secretaries who were always ready with a big hug on a hard day.

The process of writing this thesis has shown me how lucky I am in my friends.

Thank you to my roommates and best friends, Vicki and Camille and Elisa, who have been right here by my side even from thousands of miles away. Thank you to my writing buddies for keeping me on track, and to the Ladies who Lift for keeping me in shape. Thank you to the Maurauding Maidens (or is it Matrons now?) for reminding me that, if there isn't life outside the ivory tower, there are at least other ivory towers. Thank you to old Caltech friends and new Pasadena friends for pulling me out of the Caltech bubble. Thank you to my D&D group and to the Caltech and JPL softball teams for giving me a reason to return to it. Thank you to Brendan for keeping me sane, and for all the proofreading help. And thank you to my old p-set buddies and classmates and sorority sisters for making this whole experience bearable.

Above all else, I am so, so grateful to my family for their love and support. Thank you, Michelle and Michael, for making me sandwiches and filling my gas tank and pressing coffee into my hands as I run out the door yet again. Thank you, Mom, for spending hours driving me to space camp and driving me to succeed. Thank you, Dad, for teaching me to solder and helping me make my very first robot (a walking K'nex cat for the 8th grade robotics fair, where it stood out like a sore thumb next to the dioramas and drawings). I'm so glad that you're here to see this. Thank you all for your love and support for all these years.

ABSTRACT

The Jet Propulsion Laboratory and Caltech developed the Axel rover to investigate and demonstrate the potential for tethered extreme terrain mobility, such as allowing access to science targets on the steep crater walls of other planets. Tether management is a key issue for Axel and other rappelling rovers. Avoiding tether entanglement constrains the robot's valid motions to the set of outgoing and returning path pairs that are homotopic to each other. In the case of a robot on a steep slope, a motion planner must additionally ensure that this ascent-descent path pair is feasible, based on the climbing forces provided by the tether. This feasibility check relies on the taut tether configuration, which is the shortest path in the homotopy class of the ascent-descent path pair.

This dissertation presents a novel algorithm for tethered motion planning in extreme terrains, produced by combining shortest-homotopic-path algorithms from the topology and computational geometry communities with traditional graph search methods. The resulting tethered motion planning algorithm searches for this shortest path, checks for feasibility, and then generates waypoints for an ascent-descent path pair in the same homotopy class. I demonstrate the implementation of this algorithm on a Martian crater data set such as might be seen for a typical mission. By searching only for the shortest path, and ordering that search according to a heuristic, this algorithm proceeds more efficiently than previous tethered path-planning algorithms for extreme terrain.

Frictional tether-terrain interaction may cause dangerously intermittent and unstable tether obstacles, which can be categorized based on their stability. Force-balance equations from the rope physics literature provide a set of tether and terrain conditions for static equilibrium, which can be used to determine if a given tether configuration will stick to a given surface based on tether tension. By estimating the tension of Axel's tether when driving, I divide potential tether tension obstacles into the following categories: acting as obstacles, acting as non-obstacles, and hazardous intermittent obstacles where it is uncertain whether the tether would slip or stick under normal driving tension variance. This dissertation describes how to modify the obstacle map as the categorization of obstacles fluctuates, and how to alter a motion plan around the dangerous tether friction obstacles. Together, these algorithms and methods form a framework for tethered motion planning on extreme terrain.

PUBLISHED CONTENT AND CONTRIBUTIONS

- [1] M. M. Tanner, J. W. Burdick, and I. A. Nesnas, “Online motion planning for tethered robots in extreme terrain,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference On*, 2013, pp. 5557–5564. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6631375 (visited on 01/01/2014),
M.M.T drafted the algorithms, implemented the algorithms in code, collected results, and wrote the manuscript. Parts of this paper were reproduced in Section 1.5 and Section 3.5 of this dissertation, with minor edits.

TABLE OF CONTENTS

Acknowledgements	iv
Abstract	vi
Published Content and Contributions	vii
Table of Contents	vii
List of Illustrations	x
Chapter I: Introduction	1
1.1 Robotic Space Exploration	1
1.2 Potential Science Gains from Extreme Terrain Mobility	2
1.2.1 The Recurring Slope Lineae Mystery	3
1.2.2 The Moon: Moon Diver and Cold Traps	5
1.2.3 Extreme-Terrain Mobility on Earth	7
1.3 Extreme Terrain Robots	7
1.4 Other Tethered Robots	9
1.5 The Axel Rover System	12
1.6 Motivating Tethered Motion Planning	13
1.7 Literature Review	14
1.7.1 History of Tethered Motion Planning	14
1.7.2 History of Frictional Rope Mechanics	16
1.8 Thesis Contributions and Summary	16
Chapter II: A Tethered Motion-Planning Algorithm for Frictionless Surfaces	18
2.1 Motivating Scenario	18
2.1.1 Constraints	19
2.1.2 The Terrain Model	20
2.1.3 Algorithm Outline	23
2.2 Homotopy Classes	24
2.2.1 Homotopy Equivalence and Contractible Loops	25
2.2.2 Boundary Triangulated 2-Manifolds	28
2.2.3 Sleeves	31
2.2.4 Planning on a Tether-Free Plane	32
2.3 Shortest Homotopic Paths	34
2.3.1 Finding the SHP with the Funnel Algorithm	34
2.3.2 Finding all the SHPs with a Shortest Path Tree	39
2.3.3 Searching a Shortest Path Tree	40
2.4 Traversability Analysis	45
2.4.1 Checking if an Ascent is Feasible	48
2.4.2 Feasibility with a Tether Friction Cone Model	49
2.4.3 Summary: Planning on a Single Tether-Demand Plane	51
2.4.4 Multiple Planes and 3 Dimensions	52
2.5 Experimental Results	52

2.5.1	Garni Crater	53
2.5.2	Obstacle Detection	54
2.5.3	Triangulation	56
2.5.4	Path Search and Feasibility Check	62
2.5.5	Golden Queen Mine	70
2.6	Summary	78
Chapter III: Modeling and Predicting Tether-Terrain Interaction		79
3.1	Introduction	79
3.1.1	Tether Friction Obstacles	80
3.1.2	Planning with Tether Friction	81
3.2	Physics of Rope on a Smooth Frictional Terrain	82
3.2.1	Frames of Reference	83
3.2.2	Rope Model	85
3.2.3	Force Balance Equations from First Principles	86
3.2.4	Equilibrium Conditions with a Weightless Tether	87
3.2.5	Special Cases: Capstan Equation and Geodesics	88
3.3	Computing Tether Tension and Static Equilibrium	91
3.3.1	Model and Assumptions	91
3.3.2	Conditions for Sticking at a Friction Anchor	92
3.3.3	Analysis of Surface Features	94
3.3.4	Numerical Solutions for Tension	95
3.3.5	Checking if Sticking is Possible	100
3.4	Frictional Tether Planning	102
3.4.1	Assumptions about Tether Tension	102
3.4.2	The Algorithm	105
3.5	Impact of Changing Obstacles	108
3.5.1	The Pre-Planning Algorithm	109
3.5.2	Adjusting the BTM	111
3.5.3	Adjusting the Sleeve and Paths	113
3.5.4	Multiple Changed Obstacles	115
3.5.5	Update for Removed PTFOs	116
3.6	Summary and Discussion	116
3.6.1	Observations on Modeling Assumptions	117
	Risk Mitigation via a Conservative Model	117
	Operational Risk Mitigation Strategies	118
Chapter IV: Conclusion		121
4.1	Review	121
4.2	Future Work	121
4.2.1	Limitations and Assumptions	121
4.2.2	Other Suggested Improvements	123
4.2.3	Fully 3-D Surfaces	123
4.3	Applications	124
Bibliography		127

LIST OF ILLUSTRATIONS

<i>Number</i>	<i>Page</i>
1.1 Stratigraphy reveals information about the geological history of the Grand Canyon, telling a story about the region’s geography and climate during each era. Photographed by the author in 2016 from the South Rim of the Grand Canyon.	2
1.2 NASA’s Mars Exploration Rover Opportunity captured this false-color panoramic image of a wall of Victoria Crater. Loose rocks lie jumbled atop a bright band of bedrock that used to be the surface of Mars, before the crater-forming impact. Image credit: NASA/JPL/Cornell [5].	2
1.3 The dark streaks in the bottom third of the photo are Recurring Slope Lineae (RSL) in Newton Crater, Mars. Their seasonal appearance and disappearance may indicate the presence of liquid water on Mars. Image credit: NASA/JPL-Caltech/Univ. of Arizona [12].	4
1.4 Images of a possible 65-m lava tube skylight in the Marius Hills on the Moon, taken by SELENE’s cameras at different angles and times [24]. The solid white box in a) indicates an area that Haruyama et al. used for crater-counting.	6
1.5 Tethers provide support for robots on extreme terrains. In this free body diagram, a tether allows a feller buncher to log trees on steep slopes [33]. Although logging operations are not typically robotic, improved tethered planning algorithms may assist human operators or allow for eventual automation.	9
1.6 Even free-standing robots often use a tether during testing, for power or safety. Big Dog, a quadruped robot by Boston Dynamics, is shown here with a cable [46].	10
1.7 RoboBee gets power and control signals through a tether consisting of 4 tiny copper wires [47].	10
1.8 CAD rendering of the Axel rover.	12
1.9 The DuAxel rover on rough terrain. DuAxel consists of two Axel-class rovers docked with a central module.	12

1.10	Axel rappelling down a rocky slope during a field test near Black Point Lava Flow in Arizona. The tether in this image is highlighted in yellow, for visibility.	14
2.1	This Digital Elevation Map (DEM) of Andapa crater, in color, has been overlaid with a semi-transparent 25-cm-per-pixel grayscale photo to better visualize the terrain. Vertical relief is approximately 800 m. The black box delineates an example cropped area to be searched by the planning algorithm.	19
2.2	Digital Elevation Map (DEM) of an approximately 1 km-wide strip of the north rim of Andapa Crater on Mars.	21
2.3	The simplified terrain model: a 2-D tether-demand plane at angle η with polygonal obstacles. Image modified from Abad-Manterola [58].	21
2.4	An example of two paths, τ_1 and τ_2 , that are homologous but not homotopic. Image from [83].	26
2.5	An example BTM based on a figure from [65]. Here, the interior of polygon P is shaded gray, and holes in the polygon are white.	28
2.6	Part of the universal cover for the BTM in Figure 2.5.	29
2.7	An example map for a tether-demand plane. The polygon to be triangulated is outlined in black, with tether obstacles represented by the gray holes in the polygon. The starting point is at the top and the goal is at the bottom, with the gravity vector pointing downwards.	30
2.8	A boundary triangulated 2-manifold (BTM) of the same map. The obstacles are treated as holes in the manifold, and all triangulation vertices are boundary vertices.	30
2.9	The pale blue shading in these figures depicts some different possible sleeves for the same BTM. In the leftmost panel, the SHP for that sleeve is drawn as a navy blue line.	31
2.10	Labelling the triangulation diagonals, to find the crossing sequence.	33
2.11	A path with crossing sequence $\chi(\pi) = BEFGHHGGIJKLMNNO$	33
2.12	Example of a funnel in progress, inside BTM B_s . The funnel finds the shortest path $\bar{\pi}$ in sleeve $S(\pi(v_s, v_g))$, shown in pale blue. The diagonal edges that compose the sleeve are labeled in dark blue. The funnel's active edge d has left and right endpoints α and β . In this picture, $d = d_{10}$	35

- 2.13 To expand the funnel, find the vertex v opposite d_i . Label the edge incident to v that is on the sleeve as d_{i+1} . In this example, assume that the edge $\overline{\beta_i v}$ is a boundary edge. Therefore, edge $\overline{\alpha_i v}$ becomes edge d_{i+1} . Its endpoints become α_{i+1} and β_{i+1} 37
- 2.14 The predecessor of v , $t = pred(v)$, is the innermost vertex that is visible from v . Imagine extending the funnel edges as rays, λ_i , that divide the funnel into sectors. In this case, v would fall in the sector composed of the two rays intersecting t 38
- 2.15 The split operation, used when the BTM branches, turns one funnel into two. F_α is outlined in red and F_β in blue. 40
- 2.16 Using A* search with a distance metric on the Funnel Algorithm requires a modification: the relevant node being expanded for each funnel is the predecessor of v_g for that funnel. Therefore, a tether path through F_α has a key $f(F_\alpha) = g(t_g) + h(t_g)$. However, if the robot traveled via F_β , the predecessor of v_g is v . Then, $f(F_\beta) = g(v) + h(v)$. 43
- 2.17 The path and SHP from the example in Figures 2.7-2.11 are shown in green and navy blue, respectively. Intermediate anchor points are circles in purple. 46
- 2.18 The reachable set, \mathcal{R}_l , defines the limits that the robot can traverse without slipping while tied to proximal anchor a_j . It is shown in striped cyan, surrounded by dashes lines representing the reachable edges. $C_{i,j}$, the anchor reachable set for a_j , is in solid cyan. The visibility edge, dotted, extends down from a_{j-1} . Crossing this edge will release the tether from a_j , making a_{j-1} the proximal anchor. . . 47
- 2.19 A free body diagram illustrating the forces on Axel in the $x - y$ plane, at angle η to the horizontal. Not shown in the diagram is the normal force in the z -direction, $F_n = mg \cos \eta$ 49
- 2.20 In this map adapted from Abad-Manterola [58], the BTM is shown in gray and the obstacles are white. The orange arrow g shows the direction of the fall line. The SHP is in blue, with a black line representing the visibility line for the proximal anchor. Each anchor a_j has a friction cone defined by θ_j^{max} , shown in green. 50
- 2.21 DEM of the north rim of Garni Crater. The cropped selection that will be analyzed in later figures is superimposed in grayscale. 53

2.22	DEM of the cropped section on Garni's north rim, shown in color. One of the challenges in testing the obstacle detection and path planning algorithms was that most of the likely Martian terrain was relatively smooth, like this. However, that is the ideal scenario for a rover on Mars.	54
2.23	The gradient magnitude of the DEM on the left, with obstacles superimposed in color. This particular terrain seems composed of regular sandy dunes, so it is unsurprising that very few obstacles were detected.	55
2.24	The same obstacle detection algorithm was run on this 1-m resolution elevation data from Shackleton crater on the Moon. The algorithm does a much better job in an environment with sharp, clearly-defined obstacles like micro-craters and boulders.	56
2.25	The GUI used for tethered motion planning allows an operator to visually check every step of the process, from viewing the map, to triangulating, finding a path, and checking its feasibility. Buttons and checkboxes become enabled after necessary triggering events, to prevent user error.	57
2.26	The Garni Crater obstacle map resulting from Figure 2.23 is read into the program and displayed. The operator can chose the desired start and end points, depicted by the red 'x' and the green circle, respectively. Any obstacles — tiny on this map, and hard to see — are colored in blue.	58
2.27	The first triangulation step to get the BTM involves dividing the map into y-montone polygons. This map is useful for debugging, but its display can be turned off for regular use.	59
2.28	Due to the small size of the obstacles, and the large distances between them, many crater BTMs end up looking like this one: with lots of long, skinny triangles near each other.	60
2.29	A close-up of the BTM in Figure 2.28 reveals that each obstacle does, indeed, have many thing triangles connected to it. Note the scale, in meters, on the side; each of these obstacles is just a few meters wide, for a map whose resolution is 1 m.	61

2.30	This figure is primarily used for debugging the path-finding algorithm. At each step, one of the existing funnels gets overwritten or extended in a new random color. An astute reader may be able to spot some of these funnels-in-progress by their colors in this image.	62
2.31	Since obstacles were sparse in this Garni map, it is unsurprising that the resulting best (in this case, shortest) path was a straight line from start to goal, as shown in magenta.	63
2.32	The darker gray depicts the sleeve for the best path found in Figure 2.31.	64
2.33	The best, second best, and third best paths are shown in maroon. In dark gray is the sleeve for the third best path.	65
2.34	Although the sleeve in the previous image may appear to encompass an obstacle, a closer look reveals that there is actually a very small, slim BTM triangle that is not part of the sleeve.	66
2.35	An even-closer look at the obstacle in question.	67
2.36	This path is feasible; it easily falls inside the wide cyan lines, resulting from the $\mu = 0.4$ on a 45° slope.	68
2.37	Even with a smaller friction coefficient, this path is feasible too.	69
2.38	This DEM of the testing site at Golden Queen Mine is based on data from the Harvard Dataverse, by Sparky Henderson. The black rectangle represents the cropped area that was used as the robot's map.	71
2.39	The Golden Queen Mine obstacle map, as displayed by the path-planning program. As in Subsection 2.5.3, obstacles are shown in blue. An example starting point is denoted with a red 'x', and a desired goal point is denoted by the green circle.	72
2.40	The BTM for the Golden Queen Mine map.	73
2.41	Zooming in on a section of the BTM gives a better sense of the triangle structures that compose it.	74
2.42	Although it appears that the path between start and goal is a straight line, there are in fact 2 small obstacles around which the robot must detour. This can more clearly be seen in the close-up views, and in the fact that Figure 2.44 shows 2 anchors.	75
2.43	A close-up view of the final path and its sleeve.	76
2.44	The friction cones, in cyan, extend from the 2 anchors boxed in gold. Since each anchor is within the friction cone of its proximal anchor, the robot can safely ascend this path.	77
3.1	The Frenet-Serret frame for a curve $\rho(s)$ in 3-D space.	84

3.2	The Darboux frame for a curve $\rho(s)$ on a surface M	85
3.3	Ropes on a frictional sphere, illustrating the effect of ϕ on static equilibrium. When the rope is lying across the sphere's apex, $\phi = \pi$ and the rope is unlikely to slip. At a lower ϕ , however, the rope may slip.	89
3.4	Numerically computed tension along a capstan matches the well-known analytic solution, increasing or decreasing exponentially over the contact angle from 0 to $\frac{\pi}{2}$ radians.	95
3.5	Tension along a helix with a radius $R = 1$, pitch $H = 1$, and starting tension $T_0 = 45$ N.	96
3.6	A heavy rope segment (blue) lying on a titled plane, whose surface normal vector is shown in green.	97
3.7	Numerically computed rope tension for a heavy (0.1 kg/m) rope on a tilted plane.	98
3.8	A heavy tether is anchored to a shallow slope, followed by a steeper slope, in the geometry shown here.	98
3.9	The heavy tether shown will stick to the shallower slope with a low friction coefficient, but will only stick to the steeper slope at higher friction coefficients. μ also affects how quickly tension changes along the tether.	99
3.10	A small surface patch of Shackleton crater on the Moon.	100
3.11	Convexity of points on the surface.	101
3.12	Convexity of points on the surface, with arrows showing convex directions for saddle points.	101
3.13	A tether on a section of Shackleton crater. Places where sticking is possible are marked in black.	102
3.14	An example of a tether on a friction anchor, or a convex patch of terrain. Although the tether (in red) appears to go directly over the hill's apex, it is actually slightly off-center. The red vectors represent the tether normals, or Frenet-Serret normals, while the green vectors are the surface normals.	103
3.15	The angle ϕ from Figure 3.14, plotted over tether length. While the tether is at the top of the hill, ϕ is between the friction constraints, meaning that sticking is possible there.	104
3.16	Left: New obstacle added to BTM. Center: Triangulate affected area. Right: New BTM.	112

3.17 Left: New obstacle added to BTM. Right: Retriangulate affected area
for new BTM. 113

3.18 The lip of Andapa crater, from its DEM. With a slight upward slant,
it's ideal for leveraging the capstan effect to allow a weak anchor
holding force. 120

Chapter 1

INTRODUCTION

This thesis considers motion planning for rappelling tethered robots such as the Axel rover in Figure 1.10, which is described in further detail in Section 1.5. Motivated by the desire to explore hard-to-reach science targets on the Moon and Mars, Axel was designed and built to anchor the end of its tether to the top of a cliff, rappel downwards, and winch itself back up upon completion of its science goals. Such a rover, however, requires new motion planning algorithms to account for the tether's presence, ensuring that the robot can always safely descend to its science goals and then ascend to its starting position.

After describing the science goals motivating Axel's creation, this chapter discusses the issues driving the development of tethered motion planning and provides a high-level overview of tethered motion planning. Literature more specific to this work will be reviewed in the next chapter.

1.1 Robotic Space Exploration

Humankind has conquered the extremes of our planet — the summits of Everest and Kilimanjaro, the depths of the Marianas Trench, the sweltering heat of Death Valley, and the freezing stretches of Antarctica. However, despite our dominance on Earth, humans are spectacularly unsuited for spaceflight. An unfortunate astronaut, it has been estimated, can survive for 60-90 seconds without a suit [1]. A good spaceship or spacesuit can protect that astronaut for months or years, but radiation remains a major problem, especially for long-term spaceflight [1].

This is where robots shine. Robotic probes can access the furthest reaches of our solar system, with no need for a two-way ticket. Curiosity, the Mars Science Laboratory (MSL) rover, lay mostly dormant for the 9 months it took to reach Mars; it will never return [2]. The two Voyager probes have been traveling since 1977 on a grand tour of the outer solar system and now into the heliosheath and interstellar space beyond [3]. The Cassini spacecraft spent 7 years in transit to Saturn; after releasing the Huygens probe to land on Titan, it spent another 13 years investigating Saturn and its moons and rings from orbit [4].

Robots excel at the dirty, dull, and dangerous. As we move outward from Earth, to

learn about the universe and our place in it, robots can lead the way.

1.2 Potential Science Gains from Extreme Terrain Mobility

As Gary Kasparov or Ken Jennings could tell you, robots have far outpaced humans on certain scope-limited, highly computational tasks like chess or *Jeopardy!*. On ill-defined tasks, though, robots still don't hold a candle to a human child; that's why grasping, manipulation, vision, and natural language processing (to name a few) are still open research topics in the Artificial Intelligence (AI) and robotics communities. Extreme terrains pose similar problems in scope and complexity. While robotic vehicles can drive for hours on flat ground, an able-bodied human can still scramble up a rockfall that would be impossible for most robots.



Figure 1.1: Stratigraphy reveals information about the geological history of the Grand Canyon, telling a story about the region's geography and climate during each era. Photographed by the author in 2016 from the South Rim of the Grand Canyon.



Figure 1.2: NASA's Mars Exploration Rover Opportunity captured this false-color panoramic image of a wall of Victoria Crater. Loose rocks lie jumbled atop a bright band of bedrock that used to be the surface of Mars, before the crater-forming impact. Image credit: NASA/JPL/Cornell [5].

Field geologists use their natural legged mobility all the time. If you climbed down the walls of the Grand Canyon (and, ignoring the danger and illegality, you *could*), you would follow the history of the Earth down through the layers of rock. By sampling each layer, you could learn about the Earth's environment at the time, and perhaps about any organisms that lived there. Stacking those observations together to see the rock strata as a whole, you could read the region like a book. Without that ability to navigate steep slopes, though, existing extraterrestrial robots are limited to days of drilling or traversing miles of flatter ground looking for those strata.

Space agencies around the world are recognizing the importance of extreme terrain mobility. In 2012 NASA laid out the agency’s technology development needs in [6]. It states,

Extremely mobile platforms will be a critical component to both the success and diversity of extraterrestrial body exploration. . . . In chaotic unstructured environments with random geometries and large variations on [*sic*] terrain composition, robotic platforms will be challenged more than ever to avoid getting stuck while still performing relevant and meaningful exploration. . . . This technology is game changing because it provides NASA with the capability to maneuver its surface vehicles in extreme terrain in order to ‘follow the water’ — a high-priority science focus for Mars and lunar surface missions [6].

The updated 2015 version of the same document goes a step farther. Next to a picture of the Axel rover (to be described in Section 1.5), it states, “NASA needs to reach locations that may be extreme in order to find the best samples for scientific analysis. On-, above-, and below-surface mobility enables these locations to be reached with instruments for in-situ analysis or with sampling devices for sample return and more extensive analysis” [7]. The European Commission even expressed a specific interest in rappelling mobility in its recent call for space robotics technology development. They postulate that “a robot with extreme terrain mobility could explore lunar lava tubes, hunt for water ice in lunar cold traps, and investigate recurring slope lineae on Mars” [8]. The next two sections consider specific space objectives that motivate the development of Axel, and therefore this thesis.

1.2.1 The Recurring Slope Lineae Mystery

Mars is an inhospitable wasteland. Temperatures range from a frigid -100°C (-148°F) to a toasty 35°C (95°F) [9]. Worse, its approximately 700 Pa atmosphere is near the triple point of water — that is, atmospheric pressure is so low that pure water on Mars would only exist for any length of time in a solid (ice) or gaseous form [10]. This low pressure may explain why scientists have yet to find definitive evidence of liquid water on Mars, despite tantalizing hints that it once existed. One of the most suggestive indications was the discovery of Recurring Slope Lineae (RSL), first photographed in 2011 by the HiRISE camera on Mars Reconnaissance Orbiter (MRO) [11]. These long dark streaks appear and grow on sun-facing crater walls

during the Martian summers, and slowly disappear in the winter. In Newton Crater in the southern hemisphere, for example, the RSL appear on its north-facing slopes.

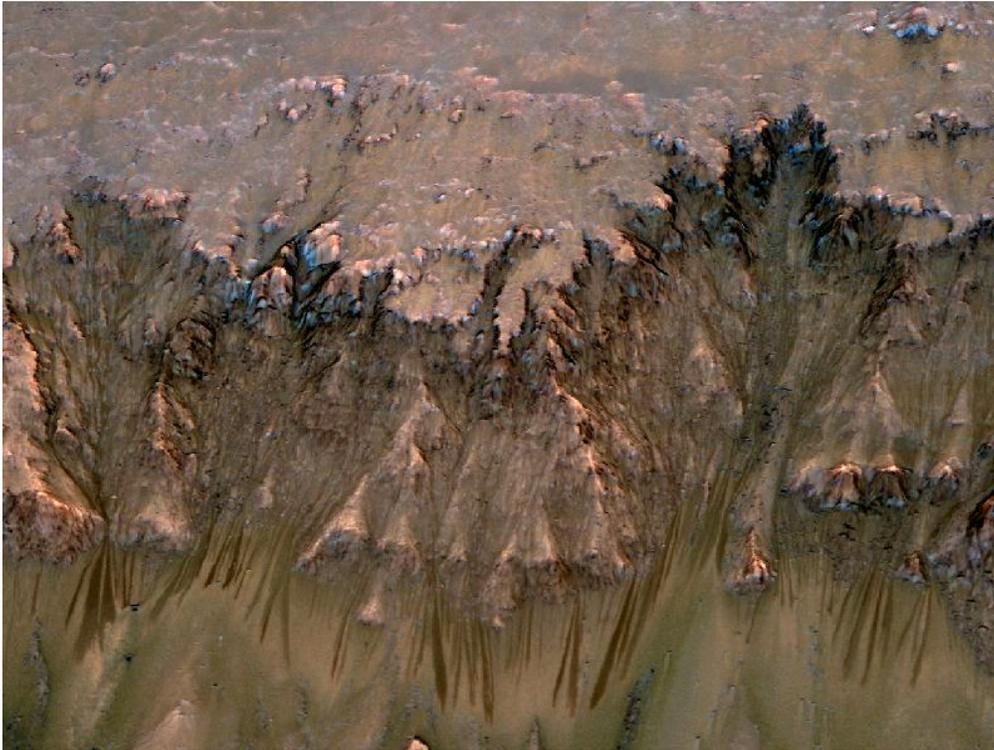


Figure 1.3: The dark streaks in the bottom third of the photo are Recurring Slope Lineae (RSL) in Newton Crater, Mars. Their seasonal appearance and disappearance may indicate the presence of liquid water on Mars. Image credit: NASA/JPL-Caltech/Univ. of Arizona [12].

The strong correlation between solar heating and the appearance of RSL streaks supports the most popular hypothesis, that RSL are caused by briny water darkening the soil [11]. As anyone who has thrown salt on an icy road knows, adding salt to water lowers its freezing point, preventing cold liquid water from freezing. Simultaneously, salt raises water's boiling point, effectively shifting the entire phase diagram down and allowing for liquid water to exist at low pressures. This hypothesis is supported by the spectral evidence of hydrated salts, which precipitate from liquid water, found by MRO's Compact Reconnaissance Imaging Spectrometer (CRISM) [13]. That same spectrometer has failed to find a spectral signature of water, but the authors point out that high-resolution spectrometry occurs in the afternoon, when the water may have evaporated too quickly to be detected. Similar debates occur over potential sources of this water [13]. Martian salt deposits are fairly common, and melting overnight frosts or water deliquescing onto those salts could form a

brine [14], [15]. However, the Martian atmosphere seems too dry for adsorption to be the mechanism [14]. Subsurface aquifers or thin layers of subsurface ice may be the source [16], and while the *Mars Odyssey* Neutron Spectrometer has not found evidence for large shallow aquifers, there may still be ice, or smaller, deeper aquifers [17], [18]. Alternately, RSL streaks might be caused by some other volatile liquid, like liquid CO₂ [19].

The lack of spectrographic evidence for water has popularized the dry granular flow hypothesis. Dundas et al. [20] explain that an avalanche could expose a darker underlying material. Over time, lighter dust would cover up that material, resulting in the streaks “disappearing”. The seasonality of the flows seems to contradict this hypothesis, but proponents have pointed out that a mostly-dry flow could still be triggered by one of the aforementioned volatile processes, and would require smaller volumes of liquid [20].

As remote sensing seems unable to resolve this debate, scientists must turn to *in situ* investigation. However, entering the craters would be a death sentence to the Mars Exploration Rovers (MER), Spirit and Opportunity, which are only designed for a 20 degree slope. The RSL occur primarily on steep, 30-40° slopes [21]. Although Curiosity is designed to withstand slopes of up to 45°, it is programmed to avoid more than 30° slopes, for safety [22]. These crater walls might have soft, slippery soil, or jumbled scree, making it all too easy to trigger a slide or rock fall. Exploring such extreme terrains would entail considerable risk for any existing rover. It is for scenarios such as these that the Axel rover, described in Section 1.5, was designed.

1.2.2 The Moon: Moon Diver and Cold Traps

The dark spots on the Moon — huge dark plains of basalts known as maria (singular: mare) after the Latin for “sea” — were formed by volcanic eruptions some 1–4 billion years ago [23]. Despite the quantity of lunar rock and soil brought back by the Apollo missions, scientists still have many questions about volcanism on the Moon. Where in the Moon’s mantle does the magma for a given mare come from, and does that source’s chemistry change over time? How hot was the flowing lava, how quickly did it flow, and how fast did it cool? What did lunar regolith look like in the past, and what was the typical time between volcanic eruptions? All of these questions are difficult to answer with only materials from the surface, possibly contaminated by billions of years of meteorite impacts and space weathering.

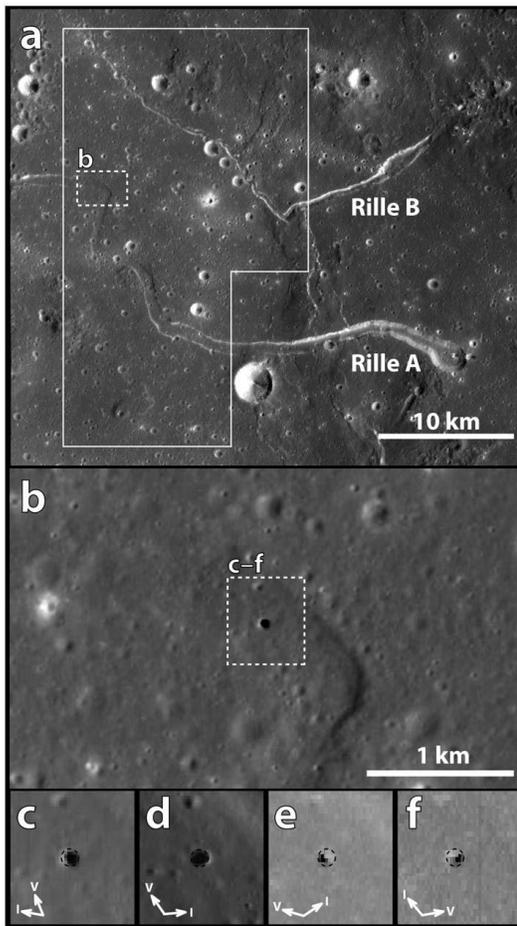


Figure 1.4: Images of a possible 65-m lava tube skylight in the Marius Hills on the Moon, taken by SELENE’s cameras at different angles and times [24]. The solid white box in a) indicates an area that Haruyama et al. used for crater-counting.

NASA’s Lunar Reconnaissance Orbiter (LRO) recently discovered deep mare pits on the Moon, in the Philolaus Crater near its north pole [25]. These pits appear to be “skylights,” which are formed when a portion of a lava tube collapses. A group of scientists from multiple aerospace centers¹ has recently proposed Moon Diver, a Discovery-class mission to investigate one of these mare pits using Axel [26]. A skylight offers a unique opportunity to investigate lunar volcanism without resorting to drilling: scientists can simply examine the stratigraphy of the layered lava composing the walls. Once the primary science objectives were met, Axel could even explore the lava tubes as a potential location for a lunar colony, since the tubes are naturally shielded from radiation.

The sun never hits its depths, making it a poetically-named “crater of eternal darkness”. As a result, the crater floor is a cold trap, one of the coldest places in the solar system. Here, too, it is possible for water ice to exist, perhaps deposited by meteorites and maintained by the bitter cold, despite the ultra-low pressure [27]. If so, this could be a valuable resource for a potential Moon colony; astronauts could mine the ice for water to drink, oxygen to breathe, and hydrogen for fuel. The Lunar Prospector spacecraft found high amounts of hydrogen in Shackleton, potentially in-

Shackleton Crater, near the Moon’s south pole, is another likely location for human habitation. Its polar location means its peaks receive near-constant sunlight, perfect for placing solar pan-

¹Scientists from: NASA Jet Propulsion Laboratory (JPL); USGS Astrogeology Science Center; DEEPS, Brown University; Johns Hopkins Applied Physics Laboratory; University of Colorado, Boulder; CEPS, Smithsonian Institution; AOPD University of Oxford, UK; NASA Marshall SFC

dicating water ice, but so far it has not been confirmed by remote sensing [28], [29]. It will, perhaps, require an extreme-terrain robot performing *in situ* measurements to confirm or disprove the presence of water.

1.2.3 Extreme-Terrain Mobility on Earth

A robot with extreme-terrain mobility would also find many uses on Earth. After the Fukushima Daichi disaster, responders sent robots into the contamination zone for reconnaissance. The robots struggled to climb stairs [30]. Improving robotic disaster response was part of the motivation behind the DARPA Robotics Challenge (DRC), which ran from 2012 to 2015 [31]. The DRC encouraged the development of robots that could semi-autonomously enter a disaster zone, ultimately aimed at helping with disaster recovery. Two of the tasks had an extreme-terrain mobility aspect: traveling through rubble and climbing stairs [31]. An extreme-terrain robot could also climb into a collapsed mining tunnel to look for survivors, or navigate the site of an avalanche without concern for a secondary slide. Historically, there have been mining disasters as a result of collapsing piles of mining waste called spoil tips. As a preventative measure, a robot could traverse those slopes to place sensors that could detect the early warning signs of a collapse in time to evacuate nearby areas. A similar system could even serve as an early warning system for California's seasonal landslides.

1.3 Extreme Terrain Robots

Even the most advanced wheeled robots can't usually climb steep slopes unassisted, and, with the exception of rocker-bogie systems, they are generally limited to climbing rocks and obstacles that are a fraction of their wheels' radius. The wheeled Curiosity rover, for example, recently failed to reach a target on a 32° slope, the steepest it had ever attempted [32]. Tracked vehicles may do slightly better, but without a tether they still struggle on steep slopes with loose surface coverings [33].

Absent the use of wheels and tracks, robots with extreme terrain mobility generally fall into 3 categories: legged robots, tether-assisted robots, and hybrid systems that involve multiple mobility modes. Noteworthy examples of legged robots include BigDog, Cheetah, LS3, Spot, Atlas, and other Boston Dynamics robots. JPL has developed Athlete, Lemur, and most recently, RoboSimian. While the technology shows promise, legged locomotion can be risky, mechanically complicated, and difficult to automate. RobotSimian and Atlas both participated in the 2015 DRC, which highlighted some of the challenges: state-of-the-art robots repeatedly fell,

damaging expensive and hard-to-replace components [34].

Dante, a tethered eight-legged walking robot developed by CMU, was one of the earliest tethered extreme-terrain robots. The first version only traveled a few feet into the crater of Mount Erebus, Antarctica, before a snapped cable ended its mission. Dante II, however, successfully explored the caldera of Mount Spurr volcano in Alaska before tipping over on the way out. A post-mortem suggested that the robot tipped over due to a combination of operator error and unforeseen forces on the tether [35], [36]. JPL's Three-Robot System for Traversing Steep Slopes (TRESSA) takes a different approach, using two anchoring robots (Anchorbots) that each spool out tether to support a third climbing robot (Cliffbot) between them [37], [38]. In recent years, single-tether climbing robots have become increasingly popular, perhaps influenced by Axel's success. Researchers from the University of Toronto have developed a prototype four-wheeled tethered robot called vScout, a retrofitted Clearpath Robotics Husky A200 [39]. vScout and its successor, Tethered Robotic Explorer (TRex), were designed primarily to help field geologists on Earth [40], [41].

The Japanese HAKUTO team was motivated by the Google Lunar XPRIZE to explore lunar lava tubes [42]. Their rover system consists of a four-wheeled rover called Moonraker, which acts as an anchor for a two-wheeled rappelling rover called Tetris. JPL's VolcanoBots 1, 2, and 3 were designed to explore volcanoes much closer to home; they have been tested at Kilauea volcano in Hawaii. Measuring only 10–12 inches across, these tiny robots are too small for an on-board spool, and are instead lowered down from above by hand [43], [44].

There are also human-operated tethered systems that may someday become autonomous, or at least more mechanized. In the timber industry, Steep Slope Machine (SSM) logging operations are highly dangerous and labor-intensive [45]. On flatter ground, loggers use a human-operated heavy machine called a feller buncher (sometimes written “feller-buncher”) to grab, cut down, and place a tree. Recently, forestry operations are experimenting with using a winch system to anchor and support the feller buncher on steep slopes, such as in the cartoon in Figure 1.5 [33]. Further mechanizing and automating the steep slope logging process may save money and human lives, but it would require an AI that can account for one or more tethers in its motion plan.



Figure 1.6: Even free-standing robots often use a tether during testing, for power or safety. Big Dog, a quadruped robot by Boston Dynamics, is shown here with a cable [46].

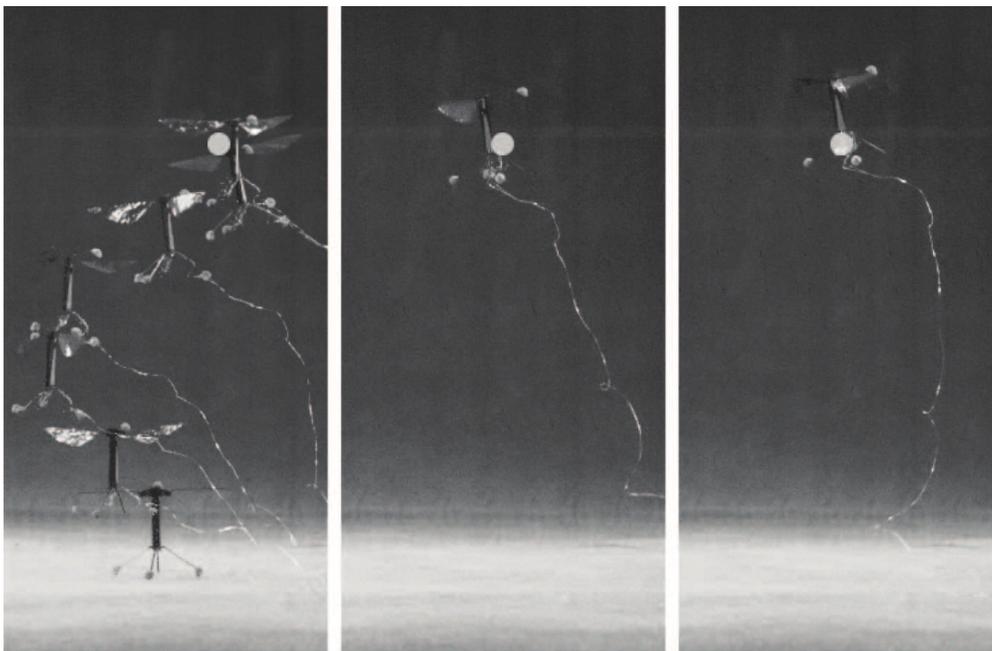


Figure 1.7: RoboBee gets power and control signals through a tether consisting of 4 tiny copper wires [47].

While radio and improving autonomy allow some workarounds to the communication problem, providing a robot with sufficient power can still pose a challenge. It is still common in the robotics community to use a power cable during lengthy testing operations, as can be seen in Figure 1.6. For RoboBee, an autonomous flying microrobot out of Harvard, the huge energy requirement of low-Reynolds-

number flight makes an on-board battery infeasible. Until that design challenge is overcome, RoboBee flies with a power cable, as shown in Figure 1.7. Quadcopters and other multirotor drones could benefit from power cables as well; a quadcopter’s high power requirement and relatively low carrying capacity limit its flight time. In fact, a few companies are already offering powered tether solutions for drones. Elistair sells the Safe-T smart tethering station for drones, and NTP sells the Powerline auto-tensioning drone system, primarily aimed at the persistent surveillance use-case [48], [49]. A startup called AeroMana is developing tethered drones to “fly for hours, not minutes” as their promotional material promises [50]. In a very different example, Ratner and McKerrow have proposed a tethered aerial robot deployed from some hovering or flying platform by a powered tether, to be used for search and rescue [51], [52]. A tether could also be used to transfer fluid, such as air pressure for a pneumatic robot.

Even untethered robots may someday need to understand the topology of tethers in order to lay a cable, pick up a leash, or unreel a garden hose. For example, robots performing repairs outside a spacecraft may need to tie themselves to the craft, like astronauts do on EVA. Window-washing robots will need safety ropes, to save pedestrians below in the event of a missed grip or broken suction cup. Each of these scenarios offers another compelling reason to research how ropes and tethers move and how to autonomously control robots that operate under the constraints of tethers.

Of course, tethered robots do have their limitations. The most obvious is the length constraint imposed by a tether, which severely reduces a robot’s mobility. Tethered robots work best in a few select situations: times when a robot need only travel a relatively short distance across environments that constrain power and communications. This is because tethers can easily become mobility-limiting, especially if they tangle or catch on obstacles. The complexity of a tether’s path can grow so as to be difficult even for a human operator, necessitating autonomous motion planning algorithms that account for a tether.

Fortunately, this thesis will show that autonomous tether planning is possible. Thus, tethered robots are ideally suited for extreme terrain or otherwise difficult-to-traverse environments, using tethers for support and safety. Tethers can also provide off-board power, freeing up a robot’s mass for locomotion. Finally, they can be useful in environments that otherwise impede communication. Axel uses its tether in all these ways: for support in abseiling down a cliff and winching itself back up again,

for communication, and for sending power to recharge its onboard batteries.

1.5 The Axel Rover System

This section was adapted and updated from [53].

In response to calls for an extreme terrain rover, researchers at the NASA Jet Propulsion Laboratory (JPL) and the California Institute of Technology (Caltech) developed the Axel class rover, and its DuAxel rover configuration. Axel is a symmetric two-wheeled robot that can rappel down steep terrain using a tether in order to carry out science missions at close range (see Figure 1.8). The tether is routed through the arm, into the tension control module, and wound around a reel in the center of its body. This tension control module, or tether management system, maintains a relatively constant tension on the tether as Axel maneuvers, and quickly picks up any slack should Axel slip.

A single Axel unit is conceived as a daughter ship in a mother-daughter configuration, with Axel's tether anchored in the mother ship (a rover or lander at the top of the extreme terrain) [54]. In addition to providing mechanical support, the tether can provide power and communication. Because the Axel body acts as a winch, paying out or reeling in the tether as it travels, the system minimizes tether abrasion.

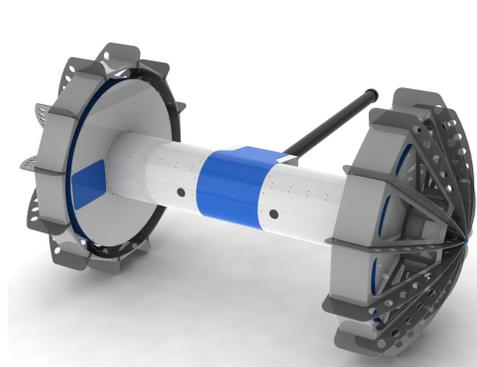


Figure 1.8: CAD rendering of the Axel rover.



Figure 1.9: The DuAxel rover on rough terrain. DuAxel consists of two Axel-class rovers docked with a central module.

Axel is minimally actuated, having a motor for each wheel, one to spool the tether, and one to control the arm through which the tether passes. This combination of motors allows Axel to rotate its body independently of its wheels, to take pictures or deploy scientific instruments from the access panels in its instrument modules while on a steep cliff face. The arm provides extra leverage for climbing over obstacles

during flat-ground traverse, and it holds the tether away from the wheels during tether-assisted climbs.

In the DuAxel configuration, two Axel-class rovers dock with a *central module*, forming a 4-wheeled rover which can traverse long distances to the edge of an extreme terrain area (see Figure 1.9), whereupon one or both Axels disengage from the central module to descend into the hazardous area, as in Figure 1.10. The central module acts as an anchor and mother ship. For more details on the geometry, mechanical design, and capabilities of the Axel and DuAxel rovers, see Matthews and Nesnas [55], Nesnas et al. [54], and Abad-Manterola et al. [56], [57]. For information on the tether tension, control, and motion planning of Axel and DuAxel, see Abad-Manterola [58], Abad-Manterola et al. [59], Tsai et al. [60], and Shankar and Burdick [61].

Axel's 4 mm-diameter custom-made tether also contains four 28-gauge copper wires wrapped around a nylon core, allowing power and communications signals to be transmitted along the tether. These components are surrounded by a VectranTM fiber strength member braid, which is wrapped in a braided black Tefzel^{®2} cover.

1.6 Motivating Tethered Motion Planning

Figure 1.10 shows Axel descending steep terrain during a field test in the Arizona desert. Clearly, without thoughtful planning of Axel's descent in such terrain, there is a high propensity for its tether to become entangled on terrain features. An operator's job is further complicated by Axel's profile; its relatively low on-board cameras (which provide perception to the remote operator) and the rough terrain mean that obstacles often occlude distant sections of slope from view. Motion planning and assistance tools would allow an operator to keep track of the tether position during complex traverses and to check that Axel can safely traverse the desired route.

Autonomy becomes even more important for extraterrestrial missions. A one-way radio signal between Earth and the Moon takes 1.25 seconds, enough delay to make teleoperation difficult. Radio signals between Earth and Mars take 4 to 24 minutes, depending on where the two planets are in their respective orbits. For this reason, safe operation on Mars necessitates at least low-level navigational autonomy, and Mars rovers are becoming increasingly autonomous over time.

²Tefzel is the brand name for ethylene tetrafluoroethylene, a fluoride-based polymer.



Figure 1.10: Axel rappelling down a rocky slope during a field test near Black Point Lava Flow in Arizona. The tether in this image is highlighted in yellow, for visibility.

1.7 Literature Review

Although tethered motion planning is a relatively new field, it draws on shortest-path algorithms from computational geometry that date back to the 1980s. Subsection 2.3.1 describes the algorithm to find the shortest path between two points in a simple polygon, as developed by Chazelle [62] and later Lee and Preparata [63]. Guibas et al. [64] developed an algorithm to create a tree of shortest paths from a point on a simple polygon to all its vertices, which is covered in Subsection 2.3.2. Hershberger and Snoeyink [65] synthesized these previous works under the framework of the universal covering space, showing that the Funnel and Shortest Path Tree algorithms can be extended to polygons with holes. For more detail, see Section 2.2. Mitchell [66] summarizes some of these results.

1.7.1 History of Tethered Motion Planning

Aside from these general shortest-path geometry solutions, most work on tethered motion planning can be categorized by the models and constraints used. We use Xavier's definitions [67] for the tether-crossing constraints, in which a *semi-planar robot* can cross its tether, a *purely planar robot* cannot, and a *pushy planar robot* can push the tether to avoid crossing it. Each tethered planning problem can be described by its combination of the following characteristics:

- Planar or 3-dimensional robot and map
- Single or multiple robots in the workspace
- Infinite (variable) or finite-length tether
- Planar, semi-planar, and push planar tether-crossing constraints

Some of the earliest works in tethered motion planning examined multiple planar point robots, some with tether length constraints. For example, Sinden [68] showed that task-scheduling for multiple tethered planar robots is equivalent to the problem of assigning edge colors for a plane embedding of a bipartite graph. He modeled the tethers as straight line segments of variable length, treating them like 2D manipulators composed of rigid links. Sinden described cases in which a solution is feasible and gave a scheduling algorithm for all robots to visit various stations on a staggered time schedule. Xavier [67] described an algorithm to find the shortest path for a single tethered robot with a semi-planar crossing constraint in $O((n_l + 1)^2 n^4)$ time, where n is the number of obstacle vertices and n_l is the number of loops in the tether's initial configuration.

Hert and Lumelsky wrote a series of papers on motion planning for multiple push planar robots with finite-length tethers in a 2-dimensional workspace. They first developed a $O(n^4)$ time algorithm to design a sequential motion strategy, where n is the number of robots [69]. It generates a relatively efficient ordering of robot motions and their trajectories to target points. Their subsequent paper [70] gave a $O(n^2 \log n)$ algorithm to find trajectories for all n robots, assuming a starting configuration and target configuration have already been found. In their third paper [71], Hert and Lumelsky represented the problem of finding an optimal target cable configuration as a Euclidean graph search with constraints. Specifically, the routes connecting the start and target configurations for each cable must not intersect. They used an exhaustive search with pruning to find a solution on this graph. Their latest paper [72] gave an algorithm for planning tangle-free motions in an open 3-dimensional workspace, such as for tethered underwater vehicles moving between stations. By modeling the tethers as straight-line segments, they analyze the intersections of the triangles formed between anchor, start point, and goal to check for triangle intersections that may indicate tangling, and use this to produce an ordering graph. They then construct a set of approximately optimal tangle-free robot paths, and compute simultaneous motion trajectories for those robots.

1.7.2 History of Frictional Rope Mechanics

Axel’s extreme-terrain purview necessitates a different approach to tethered motion planning. Although the presence of other robots and their cables are no longer a concern, the 3-dimensional terrain means that, under the right circumstances, the tether could slip over an obstacle. While Axel’s long tether is rarely a restrictive length constraint, the frictional forces on the cable between the robot and its anchor become significant. In addition, using the cable for climbing and rappelling means that any motion planning solution must take into account gravity and the robot’s ability to traverse the ground.

Sailors have been taking advantage of the mechanics of rope friction since at least the invention of the capstan or bollard. In the 1700s, Euler and Eytelwein related the holding force and the loading force of a rope wrapped around a cylinder with the eponymous Euler-Eytelwein formula, also known as the Capstan Equation. Maddocks and Keller derived the force-balance equations for a frictional rope from first principles, of which the Capstan Equation is a special case [73]. Konyukov further generalized the statics of a frictional tether to apply to an orthotropic surface, in which the friction coefficient varies based on the direction of force [74]. He also gave the nonlinear system of equations describing a rope sliding across a surface. Attaway took a more applied approach, using Coulomb’s Friction Law and the Capstan Equation to explain some common occurrences and best practice in rope rescue climbing [75]. For further detail, see Chapter 3.

1.8 Thesis Contributions and Summary

This chapter introduces the motivations for extreme terrain and tethered robots, such as the Axel rover. Axel presents some unique challenges to motion planning because it uses its tether for both climbing and stabilizing forces. While there is a body of literature on tethered motion planning, reviewed in Subsection 1.7.1, most of it is for flat ground. In his graduate work on Axel, Abad-Manterola outlined an algorithm for tethered motion planning on extreme terrain that could be modeled as a single plane [59], [58]. I have extended his algorithm to work in a wider range of circumstances, taking into account tether friction, flat planes, and the possibility of multiple planes. This thesis also describes my implementation of this algorithm, including some of the necessary details to make it work.

Chapter 2 reviews Abad-Manterola’s tethered planning work and explains the necessary background concepts of homotopy and sleeves. His algorithm started with a

simple polygon called a sleeve, found with a basic graph search. I leverage graph search techniques from the computer science community and shortest-path-finding algorithms from the computational geometry literature to make an algorithm that can more efficiently search for a useful sleeve. The graph search techniques and shortest-path algorithms are both reviewed in Section 2.3. This algorithm also produces a set of backup options in case the resulting path is infeasible, as described in Subsection 2.3.3. The resulting algorithm is described in Subsection 2.4.3. Subsections 2.2.4 and 2.4.4 explain how to apply these techniques to flatter ground or a convex series of planes. Section 2.5 gives examples of the paths resulting from applying a Python implementation of the algorithm to Martian crater data.

Chapter 3 revisits the previous chapter's assumption that there is no friction between the tether and the terrain. The resulting potential tether friction obstacles are described in Subsection 3.1.1. A review of the necessary rope physics background for the rest of the chapter is given in Section 3.2, which explains in greater detail the literature mentioned in Subsection 1.7.2. These frictional rope equations lead to a few insights about tethered robot motion and planning for tethered robots, which are demonstrated with computational examples in Section 3.3. The resulting equation that determines whether a given tether configuration will stick or slip is incorporated into the new frictional-tether path-planning algorithms in Section 3.4. One of these algorithms necessitates changes to the map structure used to hold obstacle information, and Section 3.5 describes the modifications and theory behind them.

The conclusion in Chapter 4 discusses current limitations of tethered planning, and their impact on design and operations decisions. In particular, the number of assumptions needed to model a frictional tether and the complexity of the resulting algorithm make it potentially risky for use in an expensive space mission. Fortunately, there are a number of improvements that could be made, as I discuss in Future Work. In addition, this thesis has barely scraped the surface of the literature on 3-D geodesics, which may contain exciting new solutions.

*Chapter 2***A TETHERED MOTION-PLANNING ALGORITHM FOR
FRICTIONLESS SURFACES**

This chapter starts with a description of the necessary background concepts of homotopy, BTMs, and sleeves. After a review of the literature on algorithms to find the shortest path in the sleeve, I present my improvement on one such shortest-path algorithm, in which it is adapted for graph search. This is followed by a discussion of how to conduct a feasibility analysis to ensure that a tethered robot at the bottom of a slope will be able to safely traverse an ascent path. The following section presents the results of a Python implementation of my tethered path-finding algorithm, with examples from potential mission sites in Mars craters.

2.1 Motivating Scenario

In a typical mission scenario, planners choose not a pinpoint landing location but a landing ellipse, tens of kilometers wide [76]. Since this must be a large flat area, it often means the spacecraft will land on a flat plain, some distance from the crater or geography of interest. A mobile rover, such as a DuAxel rover, must then traverse to the crater. There, DuAxel can anchor itself on the rim and separate an Axel element. Thus anchored to the top, an Axel unit can rappel down the cliff, stopping to survey the areas of interest, take samples, and perform other science tasks. If it reaches a ledge or the bottom of a crater, it may leave some slack in the tether in order to drive around without mechanical assistance. Once its tasks are completed, Axel will pull the tether taut in preparation for winching itself back up. By reeling in the tether in synchrony with turning the grousered wheels, Axel can climb back up the crater wall to its starting point [58], [61]. Upon completing its immediate science goal, Axel can dock with its mothership and begin travel to another science site.

On Mars, the Andapa, Garni, and Selevac craters have each been identified as potential landing sites for an Axel-based mission to investigate RSL [21]. Figure 2.1 shows a Digital Elevation Map (DEM) of Andapa crater, based on 1-m-per-pixel HiRISE data. Located at -4.7° latitude and -5.3° longitude on Mars' Meridiani Planum, Anadpa is of interest because of the RSL found on its northern rim. Although the crater floor is wide enough to be landable, wind-blown sand deposits form polygonal ripples that could prove treacherous to a traditional wheeled rover

approaching the crater rim from the bottom [21]. Section 2.5 will present the results of tethered path planning on Garni Crater, which is also a prime candidate for an RSL-investigation mission. It contains many RSL and is sufficiently close to the equator to be reached by current landing technology. Its relatively small crater floor is too small to be landable, so any science mission would have to travel down from the rim [21].

As Axel can only travel within a limited distance (determined by friction and slope) from the fall line¹ without slipping, the mother ship must anchor itself relatively close to the goal site in horizontal distance. Thus, we need only consider a relatively thin strip of terrain for motion planning, such as that in the black box in Figure 2.1.

2.1.1 Constraints

The objective for an autonomous motion planner is to find a descent and ascent path pair, from the start point v_s to some goal point v_g and back. Given the aforementioned motivating scenario, assume that v_s is at a higher elevation than v_g . Intuitively, if the robot travels hill right² around some rocky obstacle on the way down, it must also return on the hill right side to prevent the tether from getting wrapped around that obstacle. Maintaining this constraint for all obstacles means that the ascent and descent paths must be *homotopic*; that is, the ascent path must be smoothly deformable into the descent path and *vice*

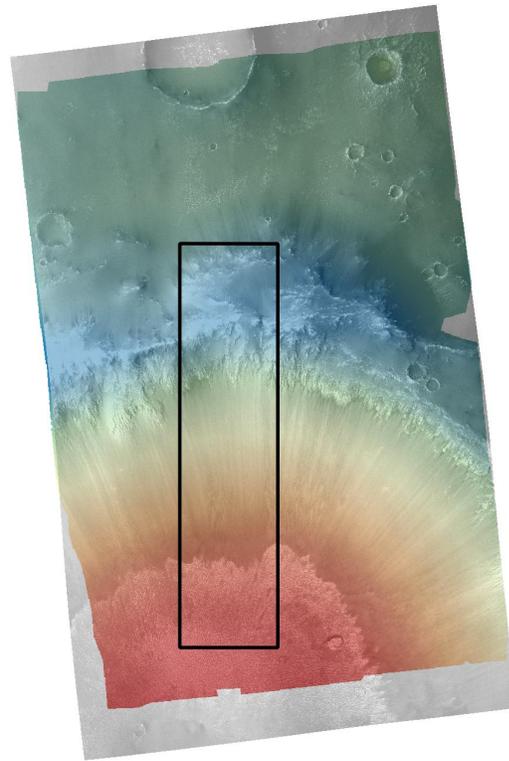


Figure 2.1: This Digital Elevation Map (DEM) of Andapa crater, in color, has been overlaid with a semi-transparent 25-cm-per-pixel grayscale photo to better visualize the terrain. Vertical relief is approximately 800 m. The black box delineates an example cropped area to be searched by the planning algorithm.

¹Fall line: a skiing term referring to the line of greatest slope, or the negative of the terrain gradient.

²Like stage right or skier's right, hill right is on the right side from the perspective of someone standing on the cliff, mountainside, or hill and facing downslope.

versa, without passing through an obstacle or moving out of the plane. For more detail, see Section 2.2. This homotopic definition does not consider the effects of friction, which may impede the actual deformation of the tether.

Having found a pair of homotopic ascent/descent paths, the robot must traverse those paths without slipping. For the descent path, this principle suggests constraining the robot to move only within a certain angle from the fall line, where that angle is determined by the friction forces that can be generated between the wheels and the crater wall. Checking the ascent path is more complicated. To ascend a tether-demand plane, the robot must first draw the tether taut. In the case of frictionless interaction between the ground and tether, the taut tether position forms the shortest path in that homotopy class. This path, the *Shortest Homotopic Path* (SHP), will be described in Section 2.3. The places where the tether contacts an obstacle will be called *anchors*. For a robot to traverse an ascent path, it must be able to move so as to remove each proximal anchor, in order, while maintaining non-slip postures. These details are discussed in Section 2.4.

2.1.2 The Terrain Model

Several assumptions will be made in modeling a candidate site's terrain. Thanks to orbiters like HiRISE, we can safely assume that the terrain is *a priori* known, at least to a low resolution. As Figure 2.2 shows, a typical crater terrain consists of sections of relatively constant slope, and can therefore be reasonably approximated as a series of n_p planes at angles $\eta_1, \dots, \eta_{n_p}$. These planes can be divided into 2 categories: *tether-free* regions, where the robot can travel without the stabilizing forces provided by its tether, and *tether-demand* regions, where Axel depends on tether forces to move [58].

This chapter describes tethered motion planning on a single tether-demand plane, which reasonably approximates most crater walls. Assume that the start point v_s lies on tether-demand plane, near the top, and that goal point v_g lies at a lower elevation on that same plane. In the motivating scenario in Section 2.1, v_g represents some science target, and v_s is at or near where Axel separated from the mothership.

The same algorithms work virtually unchanged for a series of progressively-steeper tether-demand planes which form a convex surface. In this case, an operator could select a goal point at the edge of each pair of planes, using the last v_g as the new v_s . However, a tether-demand plane followed by either a tether-free plane or a tether-demand plane at a shallower angle means that the taut tether may lift off the ground,

necessitating a more complicated 3-dimensional analysis. For further discussion, see Chapter 4.

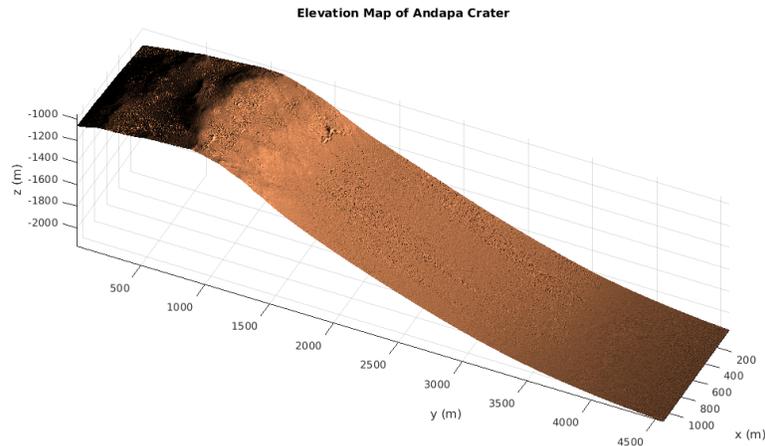


Figure 2.2: Digital Elevation Map (DEM) of an approximately 1 km-wide strip of the north rim of Andapa Crater on Mars.

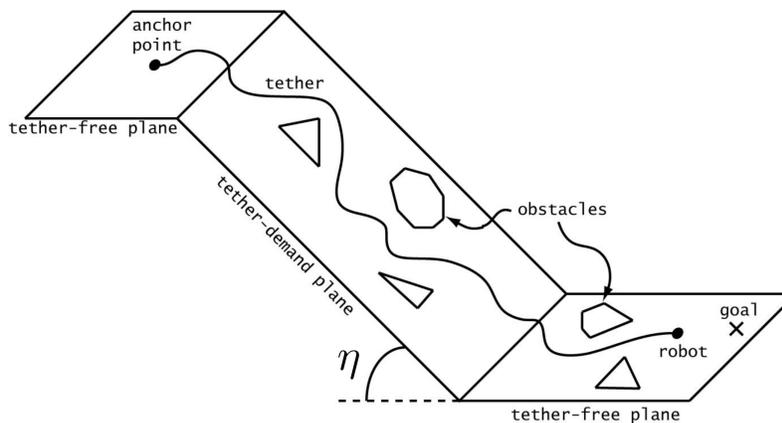


Figure 2.3: The simplified terrain model: a 2-D tether-demand plane at angle η with polygonal obstacles. Image modified from Abad-Manterola [58].

A standard traversability analysis might identify and avoid robot obstacles, boulders and rough terrain which Axel cannot traverse. However, tethered planning only concerns tether obstacles, which may differ from robot obstacles. For example, robot obstacles could include negative obstacles that sink below the plane of the ground, such as potholes and craters. These concave patches of terrain pose no

obstacle to the tether, which can pass right over them; two tether paths on either side of a negative obstacle are still homotopic to each other.

In addition, Chapter 3 will describe frictional tether obstacles, caused by a frictional tether-terrain interaction as the tether drags across the ground. Since whether a patch of ground forms a frictional tether obstacle depends on the tether configuration and the tension on the tether, these situations require further analysis. This chapter concerns only unconditional tether obstacles.

In this quasi-2-dimensional case, these tether obstacles can be modeled as a set of n_o polygons on the plane, O_1, O_2, \dots, O_{n_o} . Depending on the robot's capability and the height to which the caster arm can raise the tether, there may be positive obstacles which prove impassable to the robot but not the tether, or vice versa. In practice, the situations in which Axel can remove an anchor simply by driving over an obstacle are rare. Therefore, the following algorithms assume that tether obstacles are also robot obstacles. Since the converse is not necessarily true, a standard motion planner should be used in conjunction with the tethered motion planner, to identify robot obstacles and produce a more detailed robot path that avoids them.

Because of the sloped surface, there are a few unusual obstacles that may be difficult to model as polygons. One example is the one-way obstacle: an obstacle the robot can get down but not back up, or (more rarely) an obstacle the robot can't get down, but (starting from the bottom) could get back up. This is why in practice ascent is often harder than descent. However, by conservatively treating all such obstacles as two-way obstacles, the problem is symmetric; ascent paths work as descent paths. The problem of defining and modeling obstacles is outside the scope of this thesis, so all obstacles will be modeled as polygons here. Since the shortest path will always lie on reflex vertices,³ non-convex obstacles can be replaced in the model by their convex hull.

For the purposes of an example traversability analysis, we will assume a known, constant friction coefficient μ between Axel's wheels and the terrain. In this chapter, we will model the tether-terrain interaction as frictionless; Chapter 3 describes the changes necessary to account for tether friction.

³A reflex vertex of a polygon is concave; the angle between its incoming and outgoing edges is greater than 180° , when measured from the inside of the polygon.

2.1.3 Algorithm Outline

The aforementioned model and constraints suggest a general structure for a tethered motion planning algorithm. First, a DEM or other terrain map must be approximated as a series of planes, with tether obstacles identified using the sloped terrain obstacle-detection algorithm given in Abad-Manterola [58]. In the process, these obstacles must be further filtered to remove obstacles that sink below the plane. Assuming that a single tether-demand plane adequately captures the terrain, the following steps are taken:

- Choose a homotopy class, as represented by a sleeve.⁴
- Find the SHP in that homotopy class.⁴
- Test all the anchor points to determine if ascent is feasible. In doing so, find a traversable ascent path.
- Find a traversable descent path that is homotopic to the ascent path.

As mentioned in Subsection 2.1.2, safely-traversable ascent paths are often harder to find than descent paths. Because of this, the first steps involve finding an ascent path and checking that it can, indeed, be traversed. Since any ascent on a tether-demand plane will by definition require the mechanical assistance of the tether, the tether is drawn tight before ascent. This taut tether forms the shortest path in that homotopy class, the SHP.

Because the ascent path is so dependent on the SHP, the initial step is to find a desired SHP (or find the desired homotopy class, and then find the shortest path within it). Section 2.2 provides a primer on sleeves, homotopy classes, and a particular triangulation called the *boundary triangulated 2-manifold* (BTM), which are necessary in the rest of the chapter. Section 2.3 describes how to search for the optimal SHP, both for any criterion and specifically for the shortest SHP. Once a candidate SHP is found, Section 2.4 describes how find a feasible homotopic ascent path by checking if each anchor point is *passable*. Given a feasible ascent path, a homotopic descent path can be generated within the sleeve. If a feasible ascent path cannot be found, the next-best SHP from the the search algorithm can be analyzed.

⁴These steps can be combined, as described in Subsection 2.3.3

2.2 Homotopy Classes

Consider a topological space, X . A path π in X is a continuous function mapping the unit interval $[0, 1]$ to that topological space, denoted $\pi : [0, 1] \rightarrow X$. A path homotopy between paths π_0 and π_1 with identical endpoints is a continuous function $\Gamma(s, t) : [0, 1] \times [0, 1] \rightarrow X$ such that $\Gamma(0, t) = \pi_0(t)$ and $\Gamma(1, t) = \pi_1(t)$, and for all $s \in [0, 1]$, $\Gamma(s, 0) = \pi_0(0) = \pi_1(0)$ and $\Gamma(s, 1) = \pi_0(1) = \pi_1(1)$. The paths π_0 and π_1 are path homotopic if there exists a path homotopy between them [77], [78], [79].

Homotopy is an equivalence relation,⁵ so $\pi_0 \sim \pi_1$ will be used to denote that π_0 and π_1 are homotopic. The homotopy class of π is given by $[\pi] = \{x \in X | \pi \sim x\}$. Later, the shortest path in a homotopy class, the SHP, will be given as $\bar{\pi}$. Throughout this work, “tether” or “path” (in the context of homotopy) will be used to mean “the continuous function representing the tethers or path”, and “homotopy” will refer specifically to “path-homotopy”.

Assume a start point v_s and a goal point v_g on the surface. The descent path, π_d , has endpoints $\pi_d(0) = v_s$ and $\pi_d(1) = v_g$, and the ascent path, π_a , has endpoints $\pi_a(0) = v_g$ and $\pi_a(1) = v_s$. Define the reversal of a path π as $\tilde{\pi} = \pi(1 - t)$, i.e., the reversal goes in the direction opposite of π , from $\pi(1)$ to $\pi(0)$. Now consider two paths π_1 and π_2 with the same endpoints. One can form a loop ℓ by concatenating π_1 with the reversal of π_2 , written $\ell = \pi_1 \cdot \tilde{\pi}_2$. That loop ℓ is itself a path whose endpoints coincide at what is called the loop’s basepoint; in this case, that basepoint is at $\pi_1(0) = \pi_2(0) = \tilde{\pi}_2(1)$.

Lemma 2.2.1 *Two paths with the same endpoints, π_1 and π_2 , are homotopic if and only if $\ell = \pi_1 \cdot \tilde{\pi}_2$ is contractible [79].*

For more information on contractible loops, see Subsection 2.2.1.

With regard to tethered motion planning, the descent path, π_d , and the the reversal of the ascent path, $\tilde{\pi}_a$, must be in the same homotopy class; they start at v_s , end at v_g , and can be continuously deformed in the plane⁶ without encountering an obstacle. Similarly, $\tilde{\pi}_d$ and π_a must be in the same homotopy class. The directionality of the

⁵Equivalence relation: $\forall a, b, c \in X, a \sim a$ (reflexive); $a \sim b$ if and only if $b \sim a$ (symmetric); if $a \sim b$ and $b \sim c$ then $a \sim c$ (transitive).

⁶Unless stated otherwise, we will always be referring to homotopy in 2D. In 3D, two paths on either side of an obstacle can still be homotopic (unless a path goes through a toroidal obstacle, for example).

paths is unimportant to the theory discussed here, though, and in practice paths can easily be reversed. For the sake of succinctness, then, π_d and π_a (and any other paths with the same endpoints regardless of which is the start and which is the goal) will be referred to as homotopic to each other.

The path describing the tether when the robot is at v_g is also in the same homotopy class, regardless of whether the tether is slack or taut. In fact, the taut tether configuration forms the shortest path in that homotopy class, the SHP, which will be used as a canonical path to represent that homotopy class.

2.2.1 Homotopy Equivalence and Contractible Loops

One way of checking homotopy equivalence is to reduce the paths in question to their canonical paths, then compare those reduced paths. A canonical path can be anything that is unique to each homotopy class, and which every function in the same homotopy class can be reduced to. For example, the crossing sequence described in Subsection 2.2.3 could be used as a canonical path. Section 2.3 will cover how to find the SHP, which could also be used as a canonical path. Hershberger and Snoeyink use a path composed of segments whose vertices are the midpoint of each triangulation edge [65]. In addition, there are a number of canonical paths named in relation to obstacles.

If the number of holes, n , is small relative to the number of vertices of P , $n_h \ll n$, it becomes faster to avoid triangulation entirely. A number of related algorithms do so using trapezoidalization or decomposition, which scale with the number of holes. In general, these algorithms extend a ray or ‘fence’ from a ‘sentinel point’ chosen in each obstacle. The path is rectified, described with relation to those obstacles, and reduced. For example, Cabello et al. rectify the path by dividing it into x-monotone pieces, and assigning each a rank based on whether it lies above or below each sentinel- or end-point [80]. At this point, the path could be described by a crossing sequence through each fence [79]. Efrat et al. independently developed an algorithm based on similar principles, and later improved upon Cabello et al.’s runtime [81]. They use ‘vertical shortcuts’ to reduce a path to a sequence of x-monotone sections, in essence rectifying it. Bespamyatnikh made further improvements for handling non-simple paths [82].

The problem of whether two paths are homotopic is equivalent to the problem of whether a loop is contractible, as described in Section 2.2. If two paths with the same endpoints, π_1 and π_2 , are homotopic, then the loop ℓ formed by concatenating π_1 with

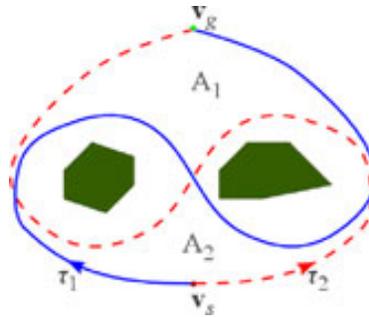


Figure 2.4: An example of two paths, τ_1 and τ_2 , that are homologous but not homotopic. Image from [83].

the reverse of π_2 is contractible. Whether the converse is true, however, depends on the definition used for homotopy, specifically with regards to the endpoints. Cabello classifies endpoints as being either ‘tacks’, ‘pins’, or ‘pushpins’ based on whether the endpoints constitute an obstacle, and if so, whether the direction in which the tether leaves that obstacle matters [80].

Technically, the Axel tethered planning problem is a hybrid tack-pushpin model. The anchoring mothership poses an obstacle to the tether, and circling around that anchor before driving off is not at all equivalent to driving off directly, so v_s should be modeled as a pushpin. The goal point, however, is imaginary; the tether or robot path may pass freely over it, so v_g can be modeled as a tack. These distinctions rarely matter in practice, though. Any reasonable motion planner will avoid encircling the start point. Similarly, if Axel were to take a path whose shortening would involve the tether path passing through the goal point, that implies that there exists a shorter path to that goal. In addition, the physics of a tether-demand plane mean that the robot will generally travel monotonically downhill.

Two paths with the same endpoints, π_1 and π_2 , are *homologous* iff $\pi_1 \cdot \tilde{\pi}_2$ forms the complete boundary of a 2-dimensional manifold embedded in X not containing or intersecting any of the obstacles [84]. Although homology is similar to homotopy, the concepts are not equivalent. If two trajectories are homotopic, they are homologous, but the converse is not true, as can be seen in Figure 2.4. As Bhattacharya et al. point out, computing homology classes is much simpler to compute, and in many robotics applications homology can be reasonably substituted [84].

Complex analysis can also be used to test the contractibility of a loop, and therefore whether two paths are homotopic. The Cauchy Integral Theorem states that if some

function $\mathcal{F}(z)$ is analytic over a simply connected region R ,

$$\oint_{\ell} \mathcal{F}(z) dz = 0. \quad (2.1)$$

By the Residue Theorem, that same contour integral around a pole will result in some non-zero multiple of $2\pi i$. Bhattacharya et al. model (x, y) nodes on a directed graph in the complex plane as $z = x + iy$ [85]. Each obstacle is represented by a pole placed somewhere inside that obstacle, ζ_j . By checking Equation (2.1), they can determine if there is an obstacle inside a given loop, and therefore whether its two constituent paths are homotopic.

Bhattacharya et al. also use the Cauchy Integral Theorem to categorize the homotopy class of a given path, π [85], [84]. They define the H-signature as

$$\mathcal{H}(\pi) = \oint_{\pi} \mathcal{F}(z) dz, \quad (2.2)$$

and

$$\mathcal{F}(z) = \frac{f_0(z)}{(z - \zeta_1)(z - \zeta_2)\dots(z - \zeta_N)}, \quad (2.3)$$

where $f_0(z)$ is any analytic function over the complex plane. Therefore, the H-signature for one path can be compared to that of another path to determine if the two paths are homotopic. If two paths (oriented in the same direction, from v_s to v_g) have the same H-signature, they are homotopic. They use homotopy as a constraint in search, allowing one to limit search to a certain homotopy class, or rule out other homotopy classes.

In several papers, Kim and his co-authors combine these approaches [86], [87], [88]. They use ray-shooting from obstacles to define a word (like a crossing sequence) that describes each homotopy class. These are placed in a homotopy-augmented graph, which can then be searched.

Tethered motion planning starts with an algorithm to find the SHP. In a terrain without obstacles, represented by a simple polygon, all paths with the same start and end point are homotopic to each other. Thus, finding the SHP is the same as finding the shortest path in that simple polygon. This can be achieved with the Funnel Algorithm, described independently by Chazelle [62] and by Lee and Preparata [63], which will be described in detail in Subsection 2.3.1. However, finding the SHP becomes non-trivial for a terrain with obstacles, represented by a polygon with holes. Hershberger and Snoeyink showed that, by using the Funnel Algorithm on the universal cover of a BTM, one can find the SHP of any path in that polygon [65].

Since this process starts with the BTM, this particular triangulation is discussed in further detail in Subsection 2.2.2, below.

2.2.2 Boundary Triangulated 2-Manifolds

A boundary triangulated 2-manifold (BTM) is a simplicial complex in which every vertex is a boundary vertex; holes are considered part of that boundary. In 2 dimensions, a simplicial complex is a collection of triangles, edges, and vertices such that any two triangles may only intersect at a vertex or at two vertices and a common edge (or else they must not intersect). Edges can have 2 incident triangles or 1, for a boundary edge. Vertices must be boundary vertices with 2 incident boundary edges, or they may lie inside a face with no incident edges [65]. For example, the vertex labeled v in Figure 2.5 lies inside the face of one of the gray triangles composing that BTM.

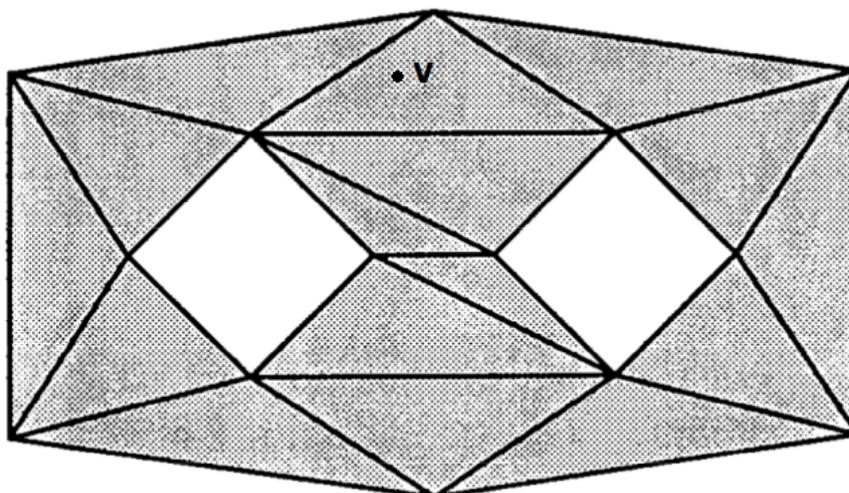


Figure 2.5: An example BTM based on a figure from [65]. Here, the interior of polygon P is shaded gray, and holes in the polygon are white.

De Berg gives an algorithm for finding a BTM, B , of a polygon with holes P in $O(n \log n)$ time, where n is the number of vertices of P .⁷ This algorithm involves partitioning P into y -monotone⁸ pieces, and then triangulating each of those pieces

⁷In fact, this algorithm applies to any planar subdivision.

⁸A monotonic function is one that is entirely non-increasing or entirely non-decreasing. A y -monotone chain of vertices is one whose y coordinates are monotonically increasing or decreasing (and an x -monotone chain has monotonically increasing/decreasing x coordinates). A monotone polygon is one composed of two monotone chains.

[89]. Note that a BTM is not unique; there can be multiple BTMs for a given polygon.

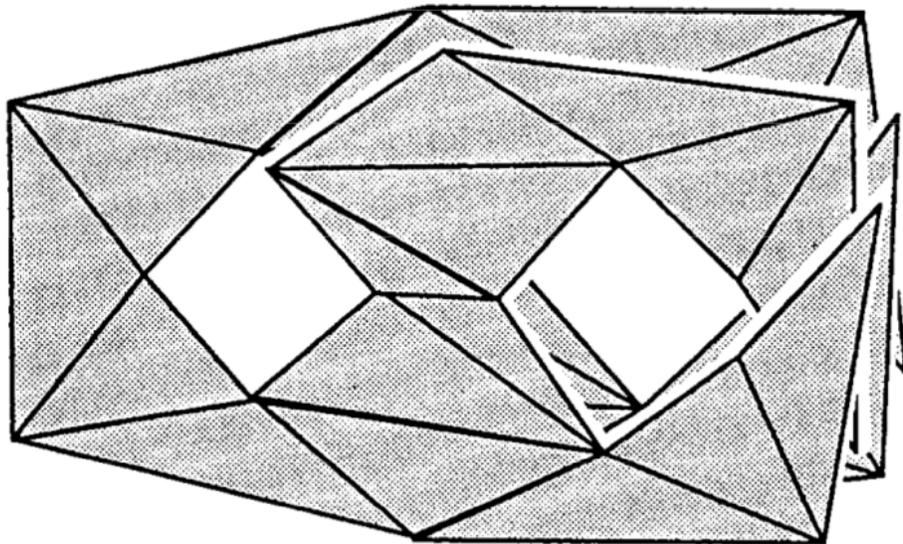


Figure 2.6: Part of the universal cover for the BTM in Figure 2.5.

Informally speaking, Hershberger and Snoeyink are “gluing” copies of each BTM triangle together to form the 3-dimensional structure representing the universal cover, as shown in Figure 2.6.⁹ Any given point p on a BTM triangle will be represented by multiple copies of that triangle in the z -axis. A roboticist might compare this to the 3-dimensional configuration space for a 2-D robot that can translate and rotate, in which a given translation of the center of that robot will be represented by multiple points with the same (x, y) coordinates but varying in z . In the configuration space, those points represent different rotations of that robot. The analogy differs in that the z -axis of the universal cover represents not robot rotations but different *ways* of reaching p , or the different homotopy classes of a path from some start point to p . Whereas one might search the configuration space for a robot motion plan, one can search this 3-D universal covering space for different homotopy classes.

⁹In short, a cover or covering space of X is a topological space \tilde{X} with a map $p : \tilde{X} \rightarrow X$ satisfying the following: there exists an open cover $\{U_x\}$ of X such that for each point $x \in X$, $p^{-1}(U_x)$ is a disjoint union of open sets in \tilde{X} , each of which is mapped homeomorphically onto U_x by p . The universal cover of X is a simple-connected covering space. For more information, see [90].

Taking the universal cover of a BTM requires starting with a base triangle, and extending the structure up or down from there. A graph representing this structure would form an infinite tree, whose root was at that base triangle. Like any tree, it can be searched. In fact, the Shortest Path Tree algorithm described by Guibas et al. [64] can be easily extended to work on a non-simple polygon, using the same universal covering space method described above. As written, this algorithm would perform a Depth-First-Search, potentially missing an easy solution as it follows one branch into infinity. However, using a queue in place of its recursive “split” call results in an effective Breadth-First Search. Subsection 2.3.2 describes this algorithm in greater detail, and Subsection 2.3.3 describes the alterations needed to turn this algorithm into a more-efficient A* search.

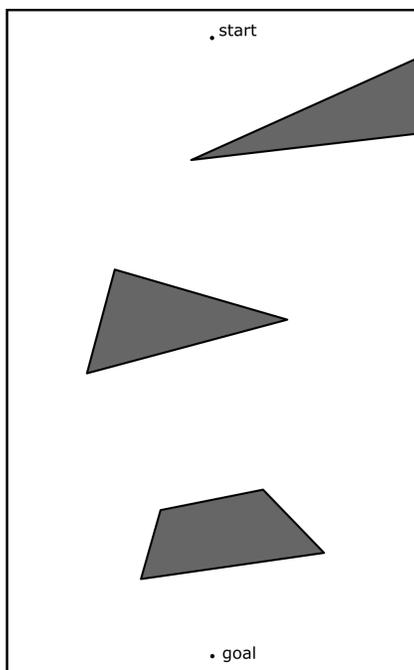


Figure 2.7: An example map for a tether-demand plane. The polygon to be triangulated is outlined in black, with tether obstacles represented by the gray holes in the polygon. The starting point is at the top and the goal is at the bottom, with the gravity vector pointing downwards.

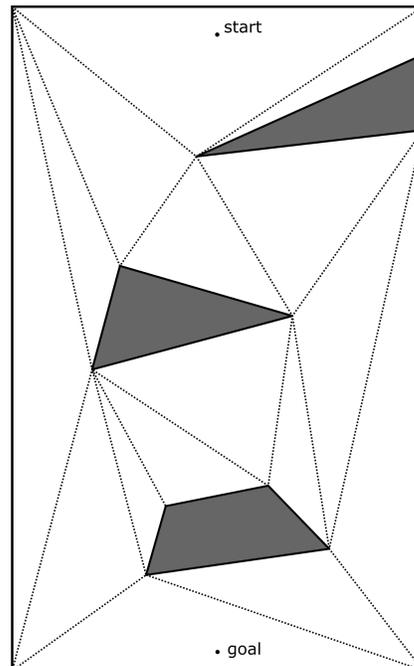


Figure 2.8: A boundary triangulated 2-manifold (BTM) of the same map. The obstacles are treated as holes in the manifold, and all triangulation vertices are boundary vertices.

The method of searching homotopy classes outlined here forms the first steps of the tethered planning algorithm. The rest of this section and Section 2.3 describe the process in greater detail. Since it forms the backbone of Hershberger and Snoeyink’s theorems, we start with that same structure, the BTM. For simplicity, we approximate the terrain as a tether-demand plane, and the polygonal tether obstacles as holes in that manifold. Note, though, that the Funnel Algorithm, Shortest Path Tree Algorithm, and my extension on the latter could be applied to any surface represented as a BTM. As long as the tether will remain on the surface, the 2D rules of homotopy apply. Thus, this process would work for any convex 3D surface that can be triangulated as a BTM.¹⁰

2.2.3 Sleeves

As stated in Section 2.2, the SHP can be used as a canonical path to represent its homotopy class. In a triangulated polygon, the series of triangles through which the SHP passes is called the *sleeve*. Lee and Preparata define a sleeve, S , as a triangulated simple polygon whose dual tree is a simple path [63]. A more intuitive definition can be found in the 3-D geodesics literature. Balasubramanian describes a *triangle chain* as a sequence of triangles such that adjacent triangles in the sequence share an edge [91]. Mitchell states that if no face appears more than once in a list of faces, that *sequence of edge-adjacent faces* is simple [92]. Thus, a sleeve is a simple triangle chain.

A sleeve can also be described by its edge sequence, the list of edges linking those

¹⁰For further discussion, see Subsection 2.4.4.

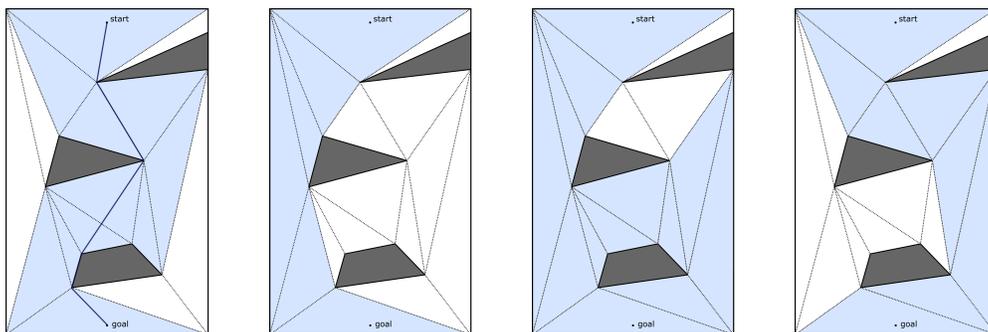


Figure 2.9: The pale blue shading in these figures depicts some different possible sleeves for the same BTM. In the leftmost panel, the SHP for that sleeve is drawn as a navy blue line.

faces. A path in a BTM, π , composed of C_π line segments has a complexity of C_π . The number of times the path crosses a triangulation edge of S is Δ_π . The sleeve containing $\bar{\pi}$, the Euclidean shortest path homotopic to π , can be found in $O(C_\pi + \Delta_\pi)$ time [65].

Because it depends on the triangulation, a sleeve is not unique to a given path. All the paths contained in a sleeve are homotopic to each other. The sleeve does not necessarily contain all the paths in a homotopy class; some paths, that exit and reenter the sleeve without encircling any holes, may also be homotopic to those within it. Still, this property of the sleeve will be used to find a descent path that is homotopic to the ascent path, by generating a descent path within the sleeve.

Let $\pi(p, q)$ be a path from p to q , and $S(\pi(p, q))$ be the sleeve corresponding to that path. To find the sleeve, start by following $\pi(p, q)$ through the BTM, noting the triangulation edges through which it passes. This edge sequence is called the crossing sequence, $\chi(\pi)$. If any edge appears more than once in $\chi(\pi)$, the crossing sequence is not simple and must be reduced to its simplest form. This reduced crossing sequence describes the sleeve. Each of these steps can be done in $O(C_\pi n)$ time, where n is the number of vertices of B , and C_π is the number of segments of $\pi(p, q)$.

To compute the crossing sequence, label the non-boundary edges of the triangulated polygon as A, B, C, etc. Every time the path passes over one of these edges, add it to $\chi(\pi)$. Then reduce that crossing sequence by removing each pair of adjacent identical letters. In the example in Figure 2.11, $\chi(\pi) = BEFGHHGGIJKLMNOPNO$ reduces to $\chi(\pi) = BEFGGGIJKLMNOPO$ and so on down to $\bar{\chi}(\pi) = BEFGIJKLMNOPO$, where $\bar{\chi}(\pi)$ is the reduced crossing sequence[79].

The sleeve S consists of all of the triangles whose common edges are given by $\bar{\chi}(\pi)$. If Δ_π is the length of the reduced crossing sequence, the sleeve can be computed in $O(\Delta_\pi)$ time.

2.2.4 Planning on a Tether-Free Plane

These concepts are all that is necessary to plan on a tether-free plane, such as flat ground. The outgoing and return paths must be homotopic to each other, but feasibility of ascent is not a concern. In fact, the tether does not need to be drawn taut, except to prevent the robot from running over it (possibly causing damage) on its return. Since Axel's tether management system ensures that the tether will maintain a constant low tension, its motion-planning on a tether-free plane consists

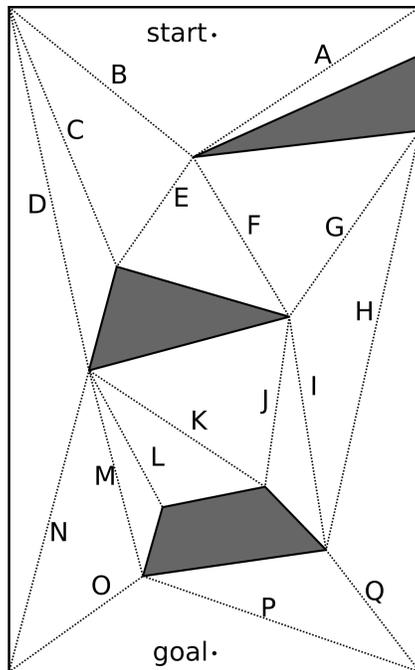


Figure 2.10: Labelling the triangulation diagonals, to find the crossing sequence.

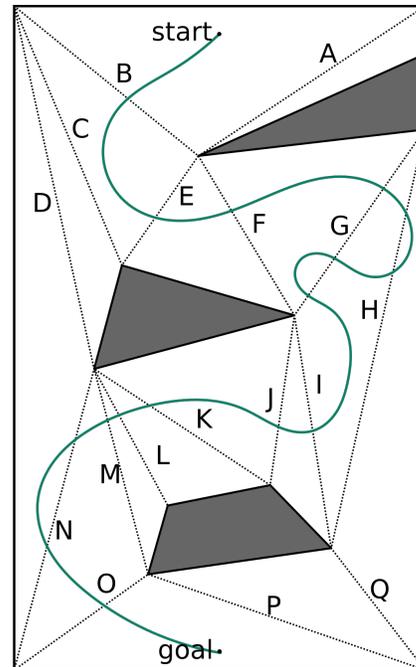


Figure 2.11: A path with crossing sequence $\chi(\pi) = BEFGHHGGIJKLMNNO$.

of the following:

1. Find a traversable outgoing path to the goal (or an incoming path from goal to start; the order doesn't matter).
2. Find its sleeve.
3. Generate another traversable path in that sleeve.

Of course, there are other ways to carry out tethered motion planning on a flat plane, as discussed in Subsection 1.7.1. The sleeve provides a nice shortcut to avoid generating a multitude of non-homotopic paths. However, if path generation is relatively cheap, one could instead generate path pairs, check if they are homotopic to each other, and continue until a homotopic pair is found.

2.3 Shortest Homotopic Paths

Given a proposed path π in a polygon P , how can one find its corresponding SHP, representing the taut tether position? The first step would be to find the BTM B by triangulating P as described in Subsection 2.2.2. Next, one would find the path's sleeve, by following π across triangulation edges to get a crossing sequence $\chi(\pi)$ and reducing that crossing sequence to its simplest form, $\bar{\chi}(\pi)$. As in Subsection 2.2.3, the sleeve $S(\pi)$ is the chain of triangles described by that edge sequence $\bar{\chi}(\pi)$.

The sleeve is a simple polygon, and the Funnel Algorithm in Subsection 2.3.1 finds the shortest path in a simple polygon. In fact, starting with a path π is not strictly necessary; one can simply find or start with a sleeve, as Abad-Manterola did, and use the Funnel Algorithm to find the shortest path in that sleeve [58]. In general, shortest-path problems are a classic topic in computer science, and can also be solved with other algorithms like visibility graphs or graph search. For this specific problem, though, the Funnel Algorithm runs the fastest. It is also important to understand because the Shortest Path Tree algorithm in Subsection 2.3.2 builds upon the same concepts.

The Funnel Algorithm only works in a single sleeve. The Shortest Path Tree Algorithm, as extended by Hershberger and Snoeyink, works on any BTM, B . This means that, rather than returning a single shortest path, it will return all the shortest paths, representing all the homotopy classes, in a BTM with holes.

2.3.1 Finding the SHP with the Funnel Algorithm

Chazelle proved that the shortest trajectory between two points in a triangulated simple polygon with n vertices can be found in $O(n)$ time, and outlined an algorithm to do so [62]. Lee and Preparata arrived independently at the same algorithm to find the shortest path in a sleeve, which forms the basis for the Funnel Algorithm below [63]. Finding the dual tree of that triangulated polygon takes $O(n)$ time, as does finding the path π through that dual tree, and therefore finding its sleeve. Therefore, the process's total running time is $O(n)$, in addition to the time needed for triangulation. Note that this is faster than using a visibility graph, which can be computed in $O(n \log n + n_E)$ for a polygon with holes, P , where n_E is the number of edges in the visibility graph [93].

Let B_s be a triangulated simple polygon with n vertices. As in Subsection 2.2, let $\bar{\pi}(p, q)$ be a chain of vertices forming the Euclidean shortest path from point p to point q in B_s . All of the vertices of $\bar{\pi}(p, q)$ (excepting the endpoints p and q) must

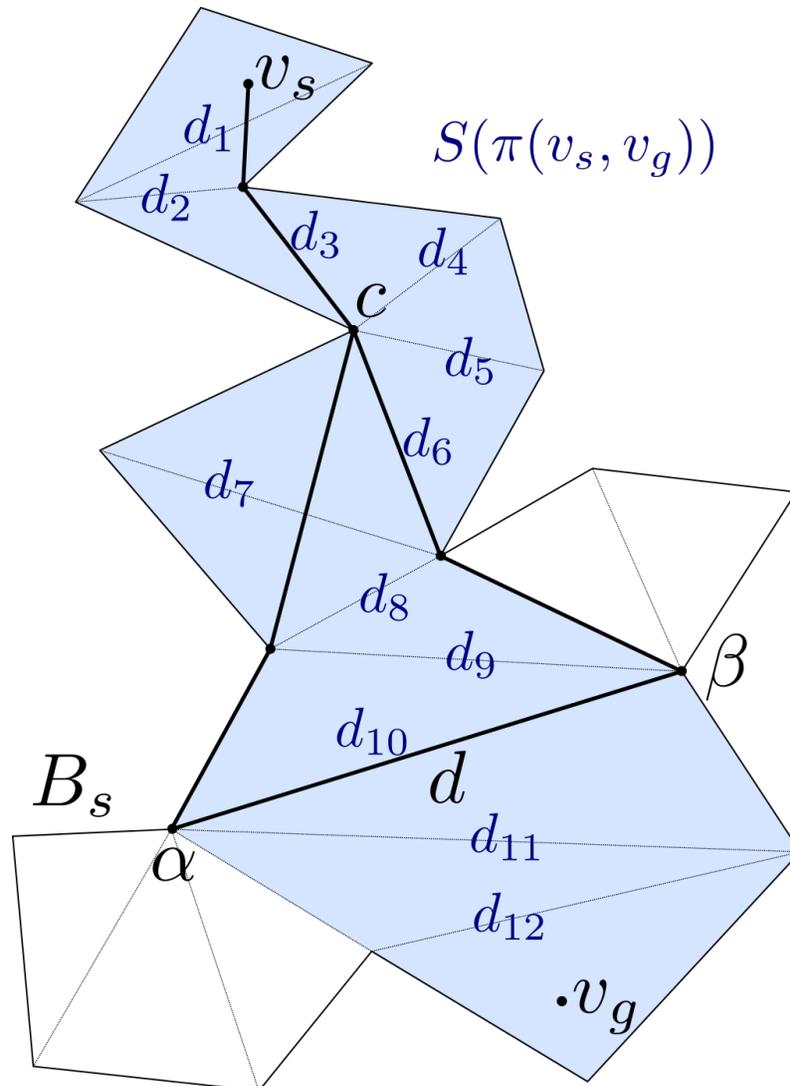


Figure 2.12: Example of a funnel in progress, inside BTM B_s . The funnel finds the shortest path $\bar{\pi}$ in sleeve $S(\pi(v_s, v_g))$, shown in pale blue. The diagonal edges that compose the sleeve are labeled in dark blue. The funnel's active edge d has left and right endpoints α and β . In this picture, $d = d_{10}$.

be reflex vertices, with an inner angle $> 180^\circ$. Assume without loss of generality that the start and goal points v_s and v_g each lie on a vertex of one of the sleeve triangles; diagonals can easily be added to make this true.

A **funnel** F is composed of the shortest paths $\bar{\pi}(v_s, \alpha)$ and $\bar{\pi}(v_s, \beta)$ from a starting point v_s to vertices α and β of B_s . Figure 2.12 shows an example funnel. The line $\overline{\alpha\beta}$, which we shall call d , forms the working edge or mouth of the funnel. Vertex c is the funnel's cusp (or apex), where $\bar{\pi}(v_s, \alpha)$ and $\bar{\pi}(v_s, \beta)$ meet; if $\bar{\pi}(v_s, \alpha)$ and $\bar{\pi}(v_s, \beta)$ share more than one vertex, those vertices will form a chain from v_s to c .

Since $\bar{\pi}(c, \alpha)$ and $\bar{\pi}(c, \beta)$ are inward convex (as a result of being the shortest paths), the region formed by this active part of the funnel must be either a triangle or convex polygon.¹¹

The Funnel Algorithm described in Algorithm 1 works by expanding and maintaining this funnel. It requires a sleeve $S(v_s, v_g)$, consisting of a series of triangles forming a chain from start point v_s to goal point v_g . The interior edges added during triangulation form the n_d diagonals d_1, d_2, \dots, d_{n_d} . The diagonals are labeled so as to increase as they move away from v_s ; in the case of planning for Axel, where v_s is above v_g , the diagonals increase going downwards.

For a diagonal d_i , let α_i refer to its left endpoint with respect to c , and let β_i be its right endpoint.¹² Note that, because diagonals may share an endpoint, sometimes $\alpha_i = \alpha_{i+1}$ or $\beta_i = \beta_{i+1}$. In Algorithm 1, γ_i will refer to whichever end vertex (α_i or β_i) is currently being expanded. For computational purposes, the funnel can be represented by a double-ended queue (or deque, pronounced “deck”), where each end of the deque is one of the active ends of the funnel, α_i and β_i .

To expand the funnel, examine vertex v that is opposite d_i on the next triangular face in the sleeve. Although v has two incident edges, only one will be a diagonal of the sleeve. (The other will either be a boundary edge, or will lead to a triangle face not in the sleeve.) That sleeve diagonal becomes d_{i+1} , as shown in Figure 2.13.

The **predecessor** of vertex v , $pred(v) = t$, is the vertex before it on the shortest path $\bar{\pi}(v_s, v)$. Thus, the predecessors form a linked list encoding the shortest path to any given point. v always has a unique predecessor. As Figure 2.14 shows, t is the innermost vertex (closest to the apex) that is visible from v . To conceptualize the predecessor, imagine dividing the funnel into sectors — triangular wedges extending to infinity — by extending a ray from each funnel vertex to its outer neighbor. If v falls into a sector, its predecessor is the funnel vertex connecting the two rays that compose that sector.

In Figure 2.14, for example, ray $\lambda_1 = \overrightarrow{v_1v_0}$, and ray $\lambda_2 = \overrightarrow{v_2v_1}$. If v fell between λ_1 and λ_2 , its predecessor would be v_1 . If v fell to the left of λ_1 , its predecessor would be v_0 . Since in this example v falls between λ_5 and λ_6 , its predecessor is v_5 .

The funnel deque is initialized with v_s and the two endpoints of the first diagonal in the sleeve. At this point, v_s is the apex of the funnel and the predecessor of all other

¹¹Guibas gives an excellent summary of the Funnel Algorithm in his lecture notes [94].

¹²The “handedness” of α and β does not matter, so long as the ordering is consistent; this definition is simply for clarity in explaining the algorithm.

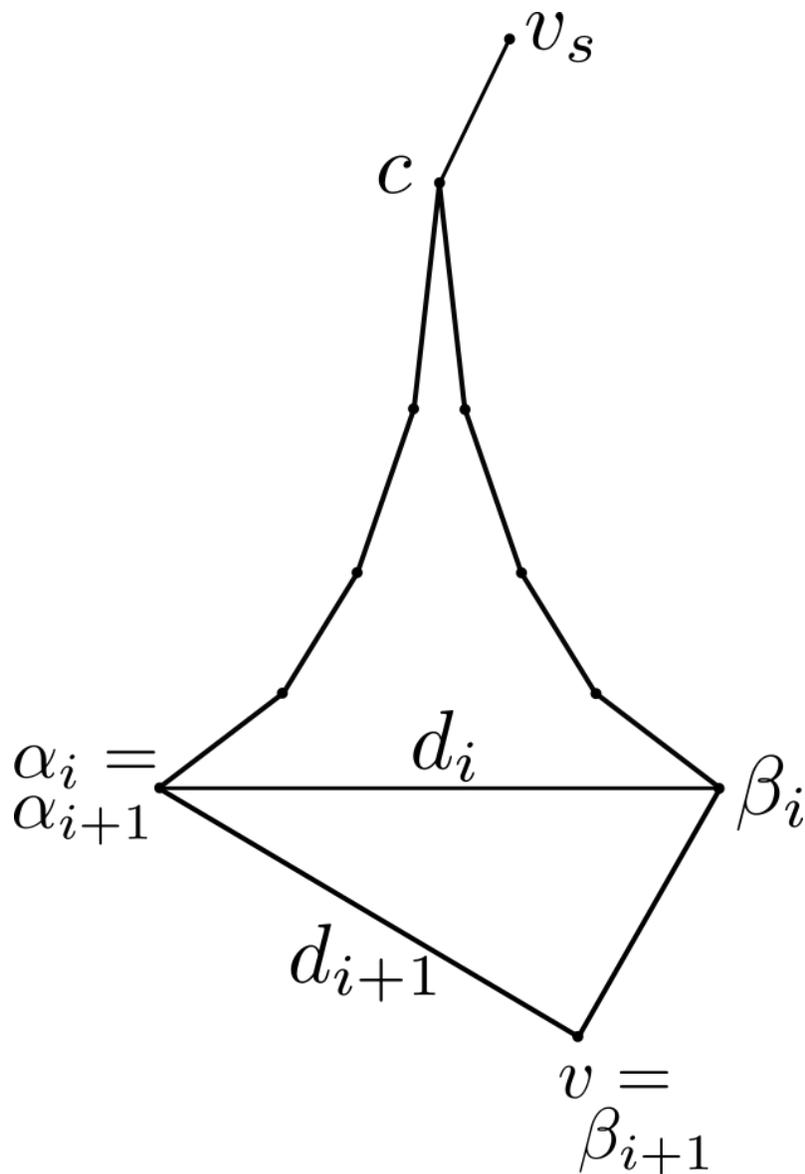


Figure 2.13: To expand the funnel, find the vertex v opposite d_i . Label the edge incident to v that is on the sleeve as d_{i+1} . In this example, assume that the edge $\beta_i v$ is a boundary edge. Therefore, edge $\alpha_i v$ becomes edge d_{i+1} . Its endpoints become α_{i+1} and β_{i+1} .

points in it. The algorithm progresses down the diagonals in the sleeve, expanding the funnel one vertex at a time. Vertex v is connected to its predecessor t , and all funnel points outside of that line \overline{vt} are discarded. Thus, the deque F increases in size with each reflex vertex, and decreases with each concave vertex. If t is past the apex, on the other side of the funnel, the funnel collapses around that point and t becomes the new apex. The process continues until the funnel reaches v_g ; if v_g

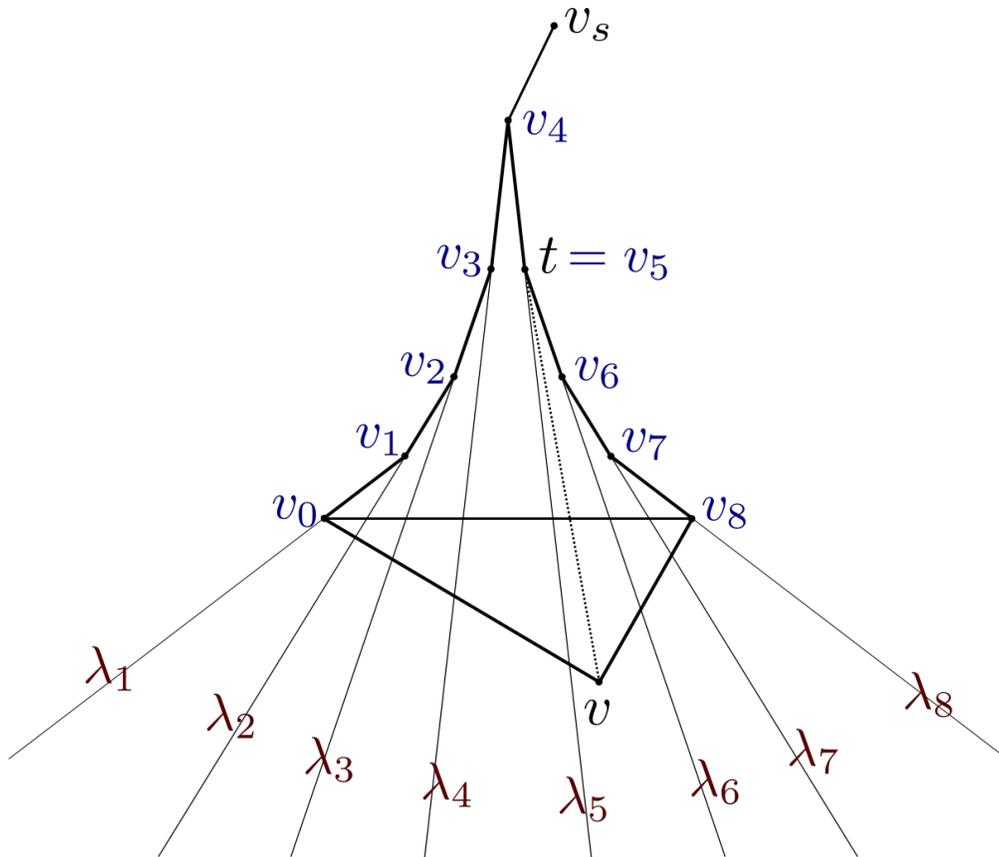


Figure 2.14: The predecessor of v , $t = \text{pred}(v)$, is the innermost vertex that is visible from v . Imagine extending the funnel edges as rays, λ_i , that divide the funnel into sectors. In this case, v would fall in the sector composed of the two rays intersecting t .

is not a vertex of the sleeve but lies inside the last triangle, an additional diagonal d_{n_d+1} should be added. At this point, the funnel should collapse down to a single path, formed by the linked list from v_g to its predecessor, and so on back to v_s . The reverse of this list forms the shortest path in the sleeve, $\bar{\pi}(v_s, v_g)$.

Algorithm 1 Funnel Algorithm

```

1: procedure FUNNEL( $v_s, v_g, S$ )
2:   Find the first diagonal,  $d_1$ 
3:   Find  $\alpha_1$  and  $\beta_1$ , the endpoints of  $d_1$ 
4:    $F \leftarrow [\alpha_1, v_s, \beta_1]$ 
5:    $pred(\alpha_1) \leftarrow pred(\beta_1) \leftarrow v_s, c \leftarrow v_s$ 
6:   for  $i \leftarrow 2$  to  $n_d + 1$  do
7:     Find the next diagonal,  $d_i$ 
8:     Find  $\gamma_i$  and its side,  $side \leftarrow \{L, R\}$ 
9:      $t \leftarrow pred(\gamma_i)$ 
10:    repeat
11:       $v \leftarrow F.pop(side)$ 
12:      if  $v == c$  then
13:         $c \leftarrow t$ 
14:      until  $v == t$ 
15:       $F.push(t, side)$ 
16:       $F.push(\gamma_i, side)$ 
17:     $\pi(v_s, v_g) \leftarrow findPath(v_g, v_s)$   $\triangleright$  Follow predecessors from  $v_g$  back to  $v_s$ .
  return  $\pi(v_s, v_g)$ 

```

2.3.2 Finding all the SHPs with a Shortest Path Tree

The *Shortest Path Tree* (SPT) gives the shortest paths from a vertex v to every other vertex on a triangulated simple polygon B_s . Lee and Preparata first gave an algorithm to find the SPT for obstacles represented by disjoint parallel line segments in $O(n \log n)$ time [63]. They handle simple polygons by choosing a sleeve and then finding the SHP in that sleeve. Guibas et al. improve upon Lee and Preparata's work, using a finger search tree data structure to allow calculation in $O(n)$ time. They extend the Funnel Algorithm to work on any triangulated simple polygon, using a recursive split operation that essentially performs a depth-first search [64].

The split operation handles the case when the dual tree splits into two or more paths. Algorithm 2, denoted as $split(d, F, c)$, requires as inputs the funnel F (represented as a deque), cusp c , and the diagonal at the current mouth of the funnel, d . If d is a polygon edge, it returns nothing. Otherwise, the algorithm finds the vertex v opposite of d , and computes $t = pred(v)$. The funnel will then be split along \overline{vt} into 2 funnels, both containing t and v . The split operation will then be called on each of these funnels in turn, with their new diagonals d_1 and d_2 .

For example, in Figure 2.15, the predecessor of v falls to the right of c , such that the original funnel is $F = [\alpha_i, \dots, c, t, \dots, \beta_i]$. After the split it becomes 2 funnels;

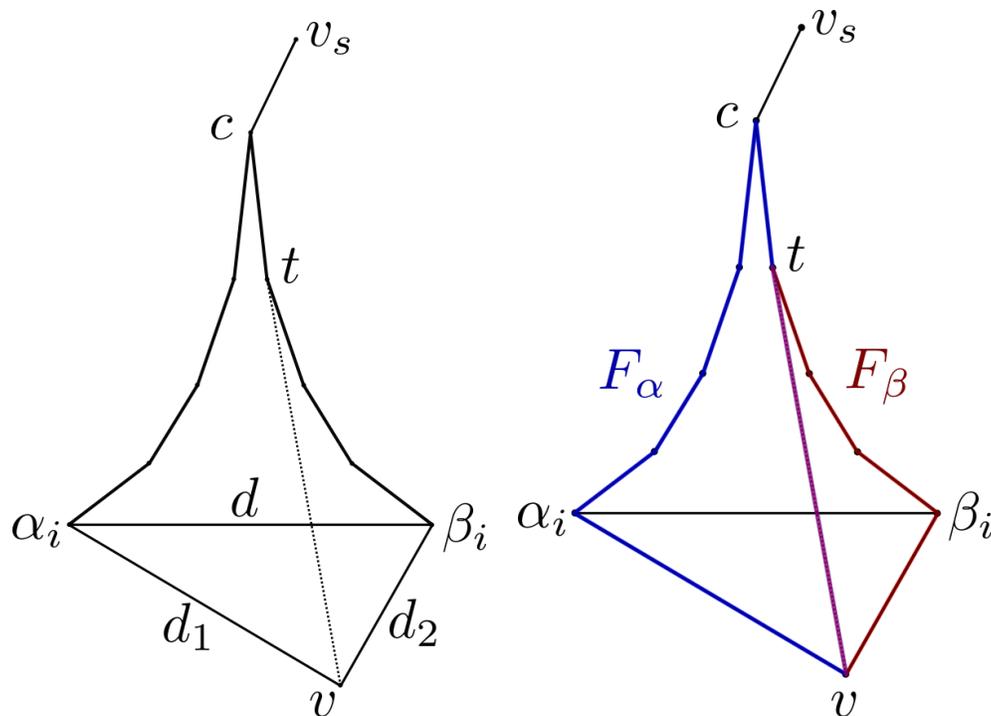


Figure 2.15: The split operation, used when the BTM branches, turns one funnel into two. F_α is outlined in red and F_β in blue.

F_α and F_β . The first funnel keeps the same cusp c , such that $F_\alpha = [\alpha_i, \dots, c, t, v]$. However, c is no longer in $F_\beta = [v, t, \dots, \beta_i]$. Instead, t becomes the new cusp for F_β . The two new split calls will be $split(d_1, F_\alpha, c)$ and $split(d_2, F_\beta, t)$. This continues recursively until the entire space has been searched.

The setup for the split algorithm is identical to that of the Funnel Algorithm. Simply replace lines 6 – 16 in Algorithm 1 with the `split()` call described in Algorithm 2.

Hershberger and Snoeyink further extend this SPT funnel algorithm to work on a polygon with holes [65]. Their innovation lies in realizing that the universal covering space of a BTM is simply connected; it has no holes. Therefore, Guibas' SPT algorithm can be applied to this lift of the BTM into the covering space, resulting in an infinite tree.

2.3.3 Searching a Shortest Path Tree

Prior to my work, only Abad-Manterola had applied homotopy class constraints to tethered rovers [58], [59]. Very few — among them, Kim, Bhattacharya, Kumar, and their collaborators — had combined search methods with homotopy [83] [85].

Algorithm 2 Shortest Path Tree Split Operation

```

1: procedure SPLIT( $d, F, c$ )
2:   Assume that  $B$  is a global variable
3:   if  $d$  is a boundary edge of  $B$  then return
4:   else
5:     Find  $v, d_1, d_2$ 
6:      $t \leftarrow \text{pred}(v)$ 
7:      $idx \leftarrow F.\text{index}(t)$ 
8:      $F_\alpha \leftarrow [F[0 : idx], v]$ 
9:      $F_\beta \leftarrow [v, F[idx : end]]$ 
10:    if  $c$  in  $F_\alpha$  then
11:       $c_1 \leftarrow c, c_2 \leftarrow t$ 
12:    else if  $c$  in  $F_\beta$  then
13:       $c_1 \leftarrow t, c_2 \leftarrow c$ 
14:    SPLIT( $d_1, F_\alpha, c_1$ )
15:    SPLIT( $d_2, F_\beta, c_2$ )

```

No one, to my knowledge, has used the Shortest Path Tree algorithm to find SHPs or homotopy class options for a tethered robot.

The Shortest Path Tree algorithm by Guibas et al. builds a tree of the shortest paths in each homotopy class [64]. This could be used to build a Shortest Path Map from a single point to every other point in a polygon. Alternately, it could be used to find all the shortest paths to a single goal.

One of my innovations was in realizing that the Shortest Path Tree algorithm was in essence carrying out a graph search, and an inefficient one at that. The recursive `split()` call is analogous to carrying out a Breadth-First Search for shortest paths (or Depth-First, depending on the order one chooses in which to make the call). The speed of that search can be improved by placing the `split()` calls in a priority queue, such that only the most likely candidates are expanded.

For tethered motion planning, that might mean optimizing for safety, perhaps by prioritizing funnels with the widest d , and therefore the highest distance between obstacles. To minimize the stress on the tether, one might discount any funnels with a high bend radius, or select those with the smallest total turning angle. Selecting funnels that are closest to the fall line would improve the chances that the best SHP is traversable. Of course, this depends on having some metric for a funnel's distance to the fall line.

Part of the difficulty in applying graph search methods to the funnel algorithm lies

in determining what constitutes a node. The gold standard in graph search, the A* algorithm prioritizes which node to visit based on an estimate of the cost of traveling to the goal via that node. In a standard implementation, the key for a node v is given by

$$f = g + h. \quad (2.4)$$

Here g is the cost to arrive at v , commonly called the cost-to-go, and h is the heuristic estimating the smallest cost to travel from v to the goal, v_g . For the search to be optimal, the heuristic must be admissible — that is, h must never overestimate the cost.

While A* can find any optimal path, its most common use is to find the shortest path. In this case, h would be a heuristic estimate of the distance from v to the goal. Euclidean or Manhattan distance from v to v_g are both commonly used distance heuristics.

Perhaps the most obvious choice of a node would be v , the next point in a split operation. This, however, would result in an inadmissible heuristic. Imagine the case in which the predecessor of the goal v_g is c . If there truly are no obstacles between the two, the taut tether would go directly from c to v_g without passing through v . The length of a path from c to v to v_g will always be equal to or longer than h , making h an overestimate of the cost. Therefore, finding the predecessor $t_g = \text{pred}(v_g)$ of the goal is crucial; t_g acts as the node, in that the shortest path to v_g is guaranteed to pass through it. Since the predecessor-finding operation depends on the funnel, t_g must be recalculated for every funnel at every step.

Consider Figure 2.16. Assume, without loss of generality, that t_g lies in funnel F_α . The shortest path to v_g through this side will pass through t_g , so the cost-to-go for F_α is $g(t_g)$. The heuristic, then, is the distance between t_g and the goal, $h = \text{dist}(t_g, v_g)$. The key is given by

$$f = g(t_g) + \text{dist}(t_g, v_g). \quad (2.5)$$

In this case any path through funnel F_β will be longer than one through F_α , since it will have to bend around v and back up towards v_g . The cost-to-go can be calculated as the cost to get to v , and the heuristic estimated as the distance from v to v_g , giving

$$f = g(v) + \text{dist}(v, v_g). \quad (2.6)$$

However, note that for F_β the predecessor of v_g is v . Therefore, Equation (2.5) works for either funnel. The SPT search is explicitly looking for the shortest path, so h

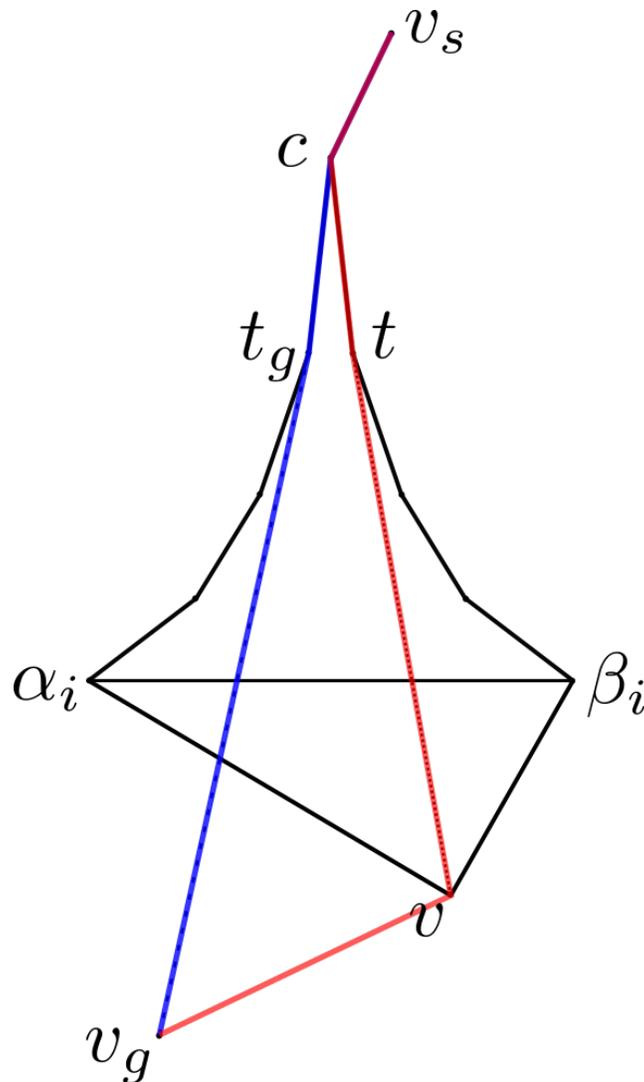


Figure 2.16: Using A* search with a distance metric on the Funnel Algorithm requires a modification: the relevant node being expanded for each funnel is the predecessor of v_g for that funnel. Therefore, a tether path through F_α has a key $f(F_\alpha) = g(t_g) + h(t_g)$. However, if the robot traveled via F_β , the predecessor of v_g is v . Then, $f(F_\beta) = g(v) + h(v)$.

can be any admissible distance metric. I have chosen to use the L2-norm, Euclidean distance, $h = \text{dist}(a, b) = \|b - a\|_2$.

The `Expand()` function does not actually return the path. However, as the graph node structure implies, one can find the shortest path to any node by tracing back that node's parent, and its parent's parent, and so on. This is what `findPath()` does, following the linked list of parent nodes from v_g back to v_s , and returning that list.

Algorithm 3 shows the setup used for searching the Shortest Path Tree; by changing

Algorithm 3 A* Search of Shortest Path Tree

Input: v_s - start point, v_g - goal point, B - triangulated polygonal map

Output: *ShpList* - list of SHPs in order of shortest to longest

```

1: procedure SPTSEARCH( $v_s, v_g, B$ )
2:   Initialize PriorityQueue  $Q$ 
3:    $ShpList \leftarrow []$ 
4:    $pathFound \leftarrow \text{False}$ 
5:   Find edges  $d_1, d_2, d_3$  of the triangle containing  $v_s$ 
6:   for  $d_i$  in  $[d_1, d_2, d_3]$  do
7:     if  $d_i$  is not a boundary edge then
8:       Find  $\alpha_1, \beta_1$  for  $d_i$ 
9:        $F \leftarrow [\alpha_1, v_s, \beta_1]$ 
10:       $pred(\alpha_1) \leftarrow pred(\beta_1) \leftarrow v_s, a \leftarrow v_s$ 
11:      Find  $v, d_\alpha, d_\beta$ 
12:       $t = pred(v)$ 
13:       $f \leftarrow g(t) + dist(t, v_g)$ 
14:       $Q.push(f, (F, v, d_\alpha, d_\beta, a))$ 
15:   while not  $Q.isEmpty()$  do
16:      $(key, data) \leftarrow Q.pop()$ 
17:      $pathFound \leftarrow \text{EXPAND}(data)$  ▷ See Algorithm 4
18:     if  $pathFound$  then ▷ Place interrupt here to avoid infinite loop
19:        $\bar{\pi} \leftarrow findPath(v_g, v_s)$ 
20:        $ShpList.append(\bar{\pi})$ 
   return  $ShpList$  ▷ List is empty if no paths found.

```

Algorithm 4 Expand Kite

```

1: procedure EXPAND( $F, v, d_\alpha, d_\beta, c$ )
2:   if  $v == v_g$  then return True
3:   else
4:      $t \leftarrow pred(v)$ 
5:      $v.g \leftarrow g(t) + dist(v, t)$  ▷ Update cost-to-go for  $v$ 
6:      $d \leftarrow [d_\alpha, d_\beta]$ 
7:     Find  $F_\alpha$  and  $F_\beta$  ▷ The same as in lines 7 – 9 of Alg. 2
8:      $F' \leftarrow [F_\alpha, F_\beta]$ 
9:     for  $i \leftarrow 0, 1$  do
10:      if  $d[i]$  is a boundary edge then
11:        do nothing ▷ Can't expand in this direction.
12:      else
13:        Find  $v', d'_\alpha, d'_\beta, c'$ 
14:         $u \leftarrow pred(v_g)$ 
15:         $f \leftarrow g(u) + dist(u, v_g)$ 
16:         $Q.push(f, (F'[i], v', d'_\alpha, d'_\beta, c'))$ 
   return False

```

the priority queue to a simple queue sans key, it can also be used with the `split()` algorithm. Algorithm 4 is analogous to `split()` in that it splits and updates the funnel, but in addition it calculates a key for each new funnel.

2.4 Traversability Analysis

One iteration of Algorithm 3 in Subsection 2.3.3 will return an SHP $\bar{\pi}_i$. The next step is to determine whether Axel can safely ascend from v_g to v_s , given this taut tether configuration. In order to ascend, Axel must be able to navigate around each obstacle that the tether contacts, in turn.

Abad-Manterola defined the **initial anchor point**, a_0 , as the location on the tether-demand plane where the tether is affixed to the host platform [58]. Assume that the n_o tether obstacles are labeled O_1, O_2, \dots, O_{n_o} . If that taut tether described by $\bar{\pi}_i$ contacts one or more tether obstacles, those points of contact are called **intermediate anchor points**[58]. Note that the tether may contact an obstacle in multiple places, or not at all, so the number of anchor points is not the same as the number of obstacles. Since an intermediate anchor point will only occur on a convex vertex of an obstacle, any anchor point analysis can be carried out on the convex hull of that obstacle, for simplicity.

Anchor points are labeled in increasing order as they proceed away from the initial anchor point. Each SHP $\bar{\pi}_i$ has a list of $n_a + 1$ anchor points (n_a intermediate anchor points and the 1 initial anchor point), $A_i = (a_0, a_1, \dots, a_{n_a})$. The j th anchor point on the i th SHP is given by $a_{i,j}$.

At any given point, the area that Axel can traverse on a tether-demand plane is determined by its proximal anchor point. Imagine the simplest case: a point robot on a plane, tied to a single initial anchor point via a tether. The area in which it could move without slipping would form a cone descending from that anchor, whose width is determined by the coefficient of friction, as described in Subsection 2.4.2. For a more complicated robot and terrain model, that traversable area is described by the **anchor reachable set** [58]. Note that anchors are determined only by contact with tether obstacles O , but a traversability analysis may involve avoiding robot obstacles as well.

Definition 2.1 *Anchor Reachable Set.* The anchor reachable set is the set of points in a tether-demand plane, $C_{i,j}(q)$, that are reachable from robot configuration q while $a_{i,j}$ is its proximal anchor point.

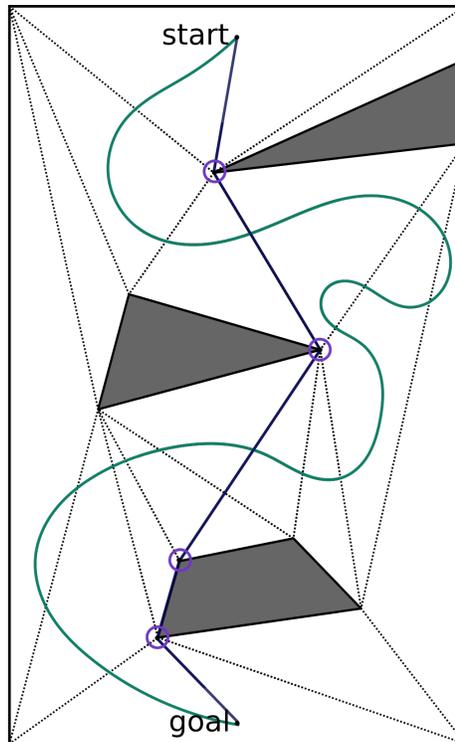


Figure 2.17: The path and SHP from the example in Figures 2.7-2.11 are shown in green and navy blue, respectively. Intermediate anchor points are circles in purple.

Clearly, the edges of the anchor reachable set could be determined by a traversability model of the robot, as in the simple friction cone model. However, this definition hints at another limiting factor for a robot on an intermediate anchor point a_j : if the robot travels far enough sideways, the tether may lose contact with the obstacle at a_j , making a_{j-1} the proximal anchor point. In this case, Axel is no longer in $C_{i,j}$ because it is in $C_{i,j-1}$. This can also be conceptualized using the predecessor from Subsection 2.3.1, which encapsulates visibility. As the robot moves sideways, a_j is no longer occluding line-of-sight to a_{j-1} .

Definition 2.2 (ARS Edges.) *The Anchor Reachable Set $C_{i,j}$ has 3 types of edges: a reachable edge, past which the robot can no longer traverse without slipping; a visibility edge¹³, beyond which a_j is no longer the proximal anchor point; and, in rare cases, an edge which may be both.*

¹³Abad-Manterola calls this the “SHP Edge”.

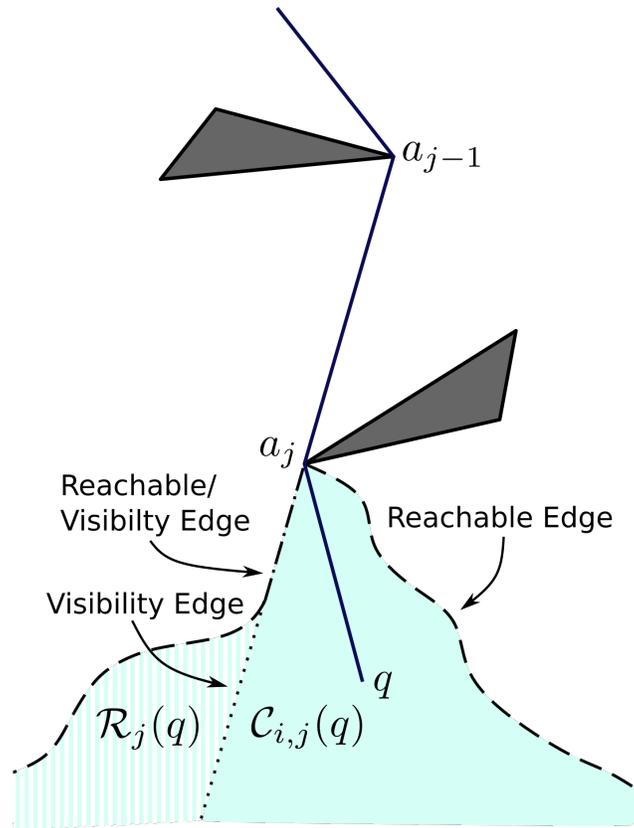


Figure 2.18: The reachable set, \mathcal{R}_j , defines the limits that the robot can traverse without slipping while tied to proximal anchor a_j . It is shown in striped cyan, surrounded by dashed lines representing the reachable edges. $\mathcal{C}_{i,j}$, the anchor reachable set for a_j , is in solid cyan. The visibility edge, dotted, extends down from a_{j-1} . Crossing this edge will release the tether from a_j , making a_{j-1} the proximal anchor.

In order to ascend past a given anchor a_j , Axel must pass over a visibility edge of $\mathcal{C}_{i,j}$, making a_{j-1} its proximal anchor. At this point, if it is not inside the reachable set for a_{j-1} , Axel will slip. From this we get the following definition:

Definition 2.3 (Passable intermediate anchor point.) An intermediate anchor point a_j is *passable* from configuration q if the robot can cross a visibility edge of $\mathcal{C}_{i,j}(q)$ to $\mathcal{C}_{i,j-1}(q)$.

To safely ascend, Axel must pass a_n , then a_{n-1} , and so on until a_1 . If it can do so, we say the ascent is feasible.

Definition 2.4 Feasibility. An SHP $\bar{\pi}_i$ or a list of anchor points A_i is *feasible* if every intermediate anchor point in turn is passable.

2.4.1 Checking if an Ascent is Feasible

Algorithm 5 below describes how to check if a proposed ascent path with a given SHP is feasible. It finds the set of all anchor points, A_i , for that SHP, then checks each for passability. To do so, it finds \mathcal{R}_j , the set of points that would be reachable if the rope were tied to a_j . L is the line of visibility, referring to the infinite ray sketched out by extending the taut tether from $A_i[j-1]$ to $A_i[j]$. \mathcal{V} refers to the set of points on the visibility edge of $C_{i,j}$, from which q' will be selected. In addition, the algorithm narrows down the selection of q' by first ensuring that any potential q' will also be inside the anchor reachable set for a_{j-1} .

Algorithm 5 Check if SHP is Feasible

Input: $\bar{\pi}_i$, a shortest homotopic path

Output: *wayPts*, if ascent feasible. False if not feasible.

```

1: procedure CHECKFEASIBILITY( $\bar{\pi}_i$ )
2:    $q \leftarrow v_g$ 
3:    $A_i \leftarrow \text{getAnchorPoints}(\bar{\pi}_i)$ 
4:    $n_a \leftarrow \text{length}(A_i) - 1$ 
5:    $\mathcal{R}_j \leftarrow \text{findTiedReachableSet}(A_i[j], q)$ 
6:    $j \leftarrow n$ 
7:    $\text{wayPts} \leftarrow []$ 
8:   while  $j > 0$  do
9:      $L \leftarrow \text{findLineOfVisibility}(A_i[j], A_i[j-1])$ 
10:     $\mathcal{V} \leftarrow L \cap \mathcal{R}_j$  ▷ Find visibility edge of  $C_{i,j}$ .
11:     $\mathcal{R}_{j-1} \leftarrow \text{findTiedReachableSet}(A_i[j-1], q)$ 
12:     $\mathcal{Q} \leftarrow \mathcal{V} \cap \mathcal{R}_{j-1}$ 
13:    while True do
14:      if  $\mathcal{Q}.\text{isEmpty}()$  then return False ▷ Ascent is not feasible.
15:      else
16:         $q' \leftarrow \mathcal{Q}.\text{pop}()$ 
17:        if  $q'$  can be reached from  $q$  then
18:           $\text{continue} \leftarrow \text{False}, q \leftarrow q'$ 
19:           $\text{wayPts.add}(q)$  ▷ This anchor point is passable.
20:         $\mathcal{R}_j \leftarrow \mathcal{R}_{j-1}$ 
21:         $j \leftarrow j - 1$ 
return  $\text{wayPts}$ 

```

In calculating the anchor reachable sets to determine if ascent is feasible for $\bar{\pi}_i$, we produce sets of reachable points and edges through which Axel must pass. There are several ways to generate ascent and descent paths from this information, depending on the traversability model and planning hierarchy in use. For example, a robot operator might simply use the visualization as a planning tool. Alternately, one could search the areas of those reachable sets to find a path, and check or ensure

that the path travels through the visibility edges in order. In Algorithm 5, we use the anchor reachable sets to develop a high-level motion plan, selecting waypoints along the visibility edges. These waypoints are sent to a second motion planner that incorporates a more detailed traversability analysis using more detailed terrain data.

2.4.2 Feasibility with a Tether Friction Cone Model

Algorithm 5 can be used with any traversability model that can determine reachable sets. The following “tether friction cone” model for a simple holonomic point robot illustrates the concept. Assume the friction coefficient μ is constant, and that the robot is on a plane with constant slope η_k .

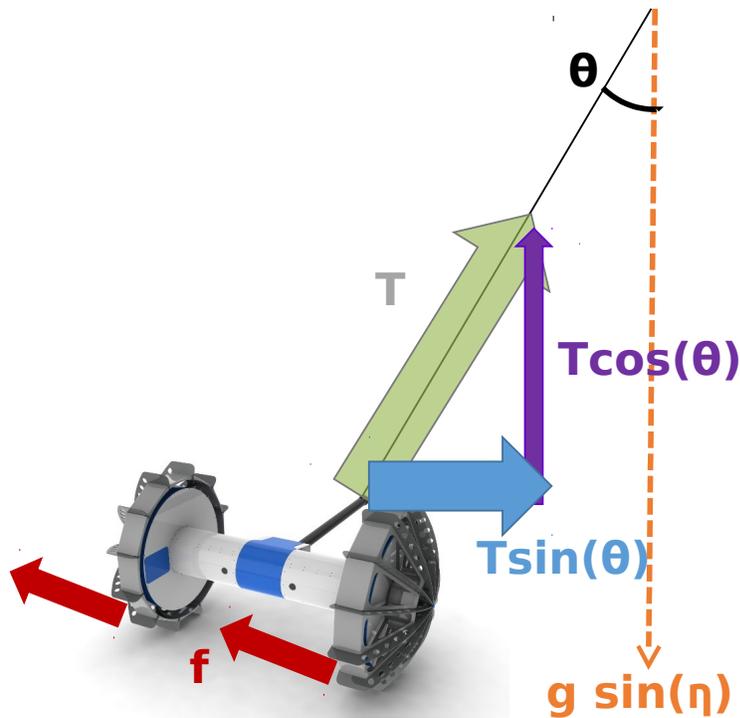


Figure 2.19: A free body diagram illustrating the forces on Axel in the $x - y$ plane, at angle η to the horizontal. Not shown in the diagram is the normal force in the z -direction, $F_n = mg \cos \eta$.

Let $\theta_j(q)$ refer to the tether angle at configuration q , with proximal anchor a_j . This refers to the angle between the taut tether and the fall line. As can be seen in Figure 2.19, the forces on the robot include tether tension T , the gravity, the normal force, and friction (which is modeled here as $f = \mu F_n$, where F_n is the normal force). Using these, Abad-Manterola [58] shows that the maximum angle the robot

can deviate from the fall line without slipping is given by

$$\left| \theta_j^{max} \right| = \sin^{-1} (\mu \cot(\eta_k)). \quad (2.7)$$

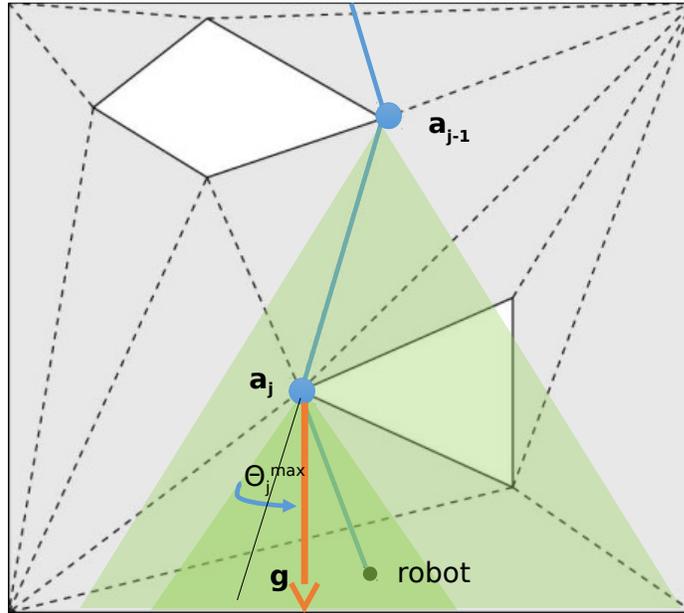


Figure 2.20: In this map adapted from Abad-Manterola [58], the BTM is shown in gray and the obstacles are white. The orange arrow g shows the direction of the fall line. The SHP is in blue, with a black line representing the visibility line for the proximal anchor. Each anchor a_j has a friction cone defined by θ_j^{max} , shown in green.

Let us call ψ_j the SHP angle, for the angle formed by the taut tether between a_{j-1} and a_j as measured with respect to the fall line. For a_j to be passable, there are 2 basic requirements:

1. $C_{i,j}$, the anchor reachable set of a_j , must have a visibility edge. For this to be true, $\theta_j^{max} > \psi_j$.
2. In order to travel from some point q on the visibility edge of $C_{i,j}$ into $C_{i,j-1}$ without slipping, a_j must be in the anchor reachable set of a_{j-1} . Therefore, $\theta_{j-1}^{max} > \psi_j$

Of course, if μ is constant throughout the surface, $\theta_j = \theta_{j-1}$, so only one of these needs to be checked.

2.4.3 Summary: Planning on a Single Tether-Demand Plane

The first step in finding an ascent/descent path pair is to preprocess the data. This involves approximating terrain data as a series of planes, and finding the obstacles on each plane k represented by polygon P_k . Then the space around those obstacles is triangulated into a BTM. For each triangulated plane that is a tether-demand plane, Algorithm 6 can be run to find a feasible descent and ascent path.

Algorithm 6 summarizes the process of finding the shortest (or best) SHP via A* SPT search, checking that SHP for feasibility, and using waypoints from that feasibility check to find an ascent path. The descent path could also be generated with those same waypoints. Alternately, one could generate a homotopic descent path in the sleeve and check that it is traversable. If at any point a path cannot be found, the process can be repeated for the next-best SHP.

Because there are many different options for how to find the paths, each best-suited for a different environment, Algorithm 6 has left the details vague.

Algorithm 6 Find a Feasible Ascent/Descent Path Pair

Input: start v_s , goal v_g , polygonal map P of tether obstacles on a plane

Output: (π_a, π_d) ascent and descent path pair

```

1: procedure PATH PLANNING( $v_s, v_g, P$ )
2:    $B \leftarrow \text{findBTM}(P)$ 
3:    $searcher \leftarrow \text{SPTSEARCH}(v_s, v_g, B)$  ▷ Initialize search
4:   while True do ▷ Or while ( $count < cutoff$ )
5:      $\bar{\pi} \leftarrow searcher.getNextBest()$ 
6:      $wayPts \leftarrow \text{CHECKIFFEASIBLE}(\bar{\pi})$ 
7:     if  $wayPts$  is not False then
8:       find ascent path  $\pi_a$  through  $wayPts$ 
9:       find corresponding descent path  $\pi_d$ 
10:    return  $(\pi_a, \pi_d)$ 

```

Note that Algorithm 6 has the potential to loop forever. Any given BTM with one or more obstacles has infinite homotopy classes, since a path can always loop around an obstacle (or several obstacles) more times. This could potentially be solved with a loop-check in Algorithm 3 to ensure that only simple paths are returned. However, that SPT search might still return paths that, while in unique homotopy classes, would not be well-suited for Axel to travel. For example, any path that requires the robot to move upwards from an anchor, unsupported by its tether, would not be traversable. While these paths will naturally be rejected during the Algorithm 5 feasibility check, they will also have a longer travel distance, and thus be returned

later in the search. For these reasons, in practice an operator can quickly determine when Algorithm 6 has ceased to return any useful paths, and terminate the search at that time.

2.4.4 Multiple Planes and 3 Dimensions

Recall that a sleeve is the same as a ‘triangle chain’. Balasubramanian et al. state, “it is possible to isometrically flatten any triangle chain of a triangular mesh... [b]y isometrically flattening the triangle chain, we also map the curve to the plane isometrically. Under such a mapping, the intrinsic geometry of the surface is preserved” [91]. Because of this, the Funnel Algorithm or its variants can still be used on any BTM with holes, even if that BTM consists of several planes in a row. In addition, the tether will only remain in contact with the surface as long as the surface is convex. Therefore, planning can be safely carried out for any series of planes where for each plane k , $\eta_k > \eta_{k-1}$.

As long as the tether remains in contact with the surface, most of Algorithm 6 remains the same for planning on multiple planes. The planner must still find an SHP, check that ascent is feasible (for any tether-demand plane), and then generate homotopic ascent and descent paths in the sleeve. Only the traversability model depends on the plane, affecting the anchor reachable sets. In addition, the tether must be drawn taut before moving from any tether-free plane to a tether-demand plane.

It should be possible to find a shortest path or SHP in a more complicated polyhedral surface, without approximating it as a series of planes. There are several algorithms for finding geodesics or shortest paths on polyhedral surfaces [91], [92], [95]. Many of these algorithms are slow for large numbers of vertices, so a number of faster approximation algorithms have also been developed [96]. However, a geodesic is only a locally shortest path; they may differ from globally shortest paths, especially around boundaries. Balasubramanian et al. present an algorithm for finding shortest paths [91], but their approach provides paths that lie on the surface, even in the presence of any concavities. Because a taut tether would lose contact with the surface on these concavities, their algorithm would need to be modified to handle terrain concavities.

2.5 Experimental Results

To demonstrate the utility of this tethered path-finding algorithm, I wrote a program that analyzed real-world data for potential paths. The following subsections show

results for data sets from Garni Crater on Mars and Golden Queen Mine on Earth.

2.5.1 Garni Crater

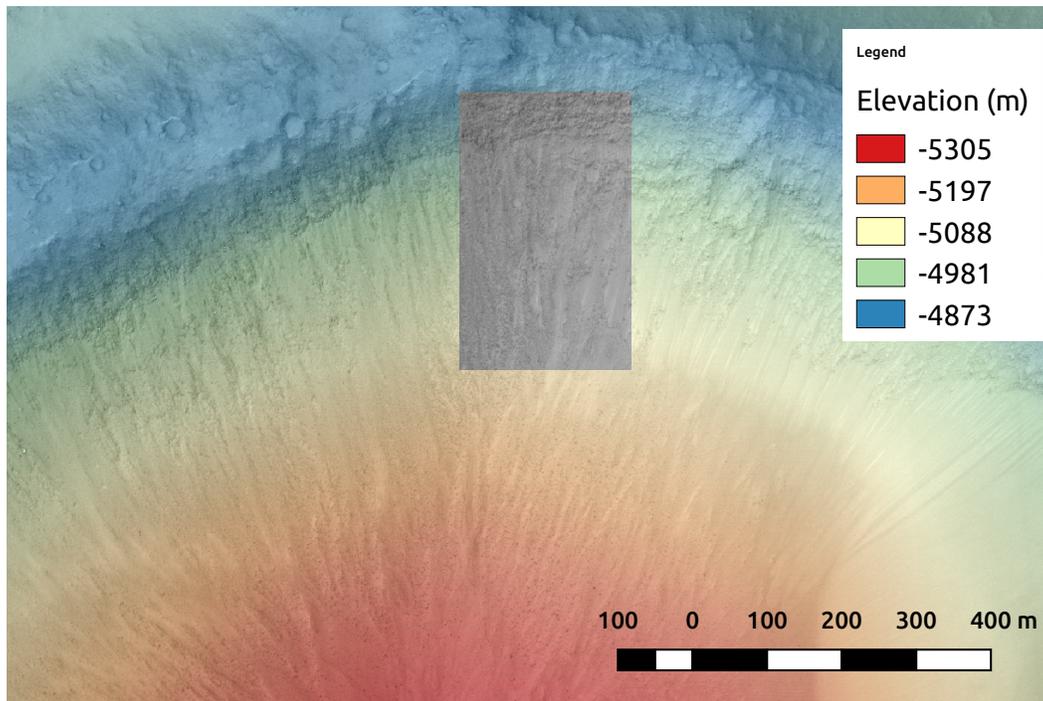


Figure 2.21: DEM of the north rim of Garni Crater. The cropped selection that will be analyzed in later figures is superimposed in grayscale.

As Section 2.1 mentioned, Garni Crater is one likely mission destination for Axel. The DEM shown in Figure 2.21 was based on data taken by the MRO HiRISE camera, with a resolution of 1 m per pixel in x and y . The Geographic Information Systems (GIS) program QGIS was used to produce Figure 2.21, combining the DEM (in color) with a grayscale semi-transparent photo for texture. QGIS was also used to crop a strip of the crater's north rim that has a relatively constant slope, shown in grayscale in Figure 2.21 and in color in Figure 2.22. This cropped strip, as seen in Figure 2.22, will be modeled as a single tether-demand plane in the rest of this section.

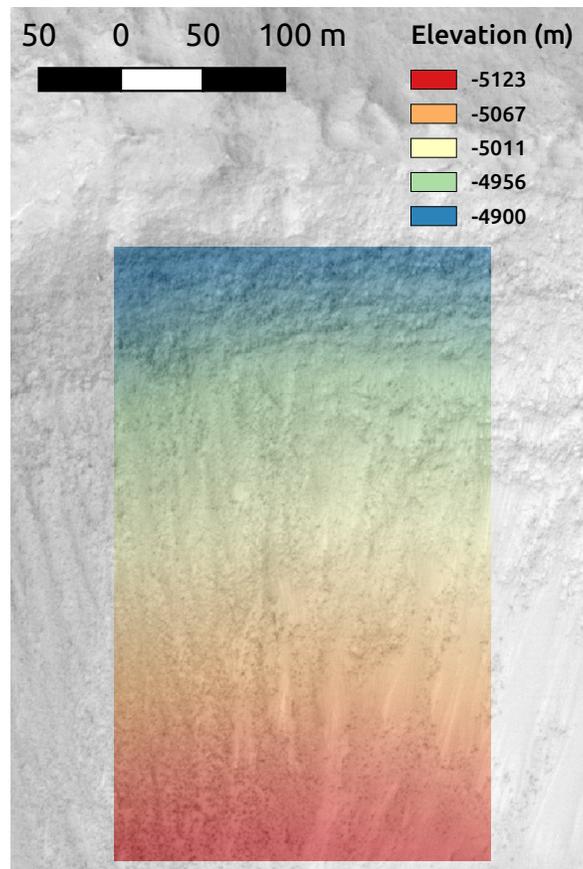


Figure 2.22: DEM of the cropped section on Garni's north rim, shown in color. One of the challenges in testing the obstacle detection and path planning algorithms was that most of the likely Martian terrain was relatively smooth, like this. However, that is the ideal scenario for a rover on Mars.

2.5.2 Obstacle Detection

The cropped selection was then read into Matlab for obstacle detection. I wrote a program to do so based on Abad-Manterola's algorithm for obstacle detection on a slope [58], which in turn was based on the extreme-terrain algorithm developed by Manduchi et al. [97]. These algorithms use the parameters H_{min} , H_{max} , and θ_{max} , which depend on the vehicle's ground clearance, the vehicle's width, the quality of the data, and the maximum slant the vehicle can safely traverse.¹⁴ In the following examples, I used $H_{min} = 0.2$ m, $H_{max} = 2$ m, and $\theta_{max} = 40^\circ$. Figure 2.23 depicts the resulting detected obstacles as colored polygons, which are superimposed on a grayscale image of Garni's gradient magnitude. The smoothness of Garni's crater wall at this location makes it difficult to gauge the obstacle detection's effectiveness. However, a crater-littered patch of Shackleton crater on the Moon provides a target-

¹⁴Tailoring the parameter values to the situation is an inexact science. For further discussion on how to choose them, see [97].

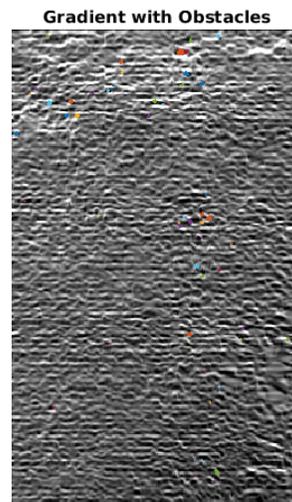


Figure 2.23: The gradient magnitude of the DEM on the left, with obstacles superimposed in color. This particular terrain seems composed of regular sandy dunes, so it is unsurprising that very few obstacles were detected.

rich test environment, as can be seen in Figure 2.24.

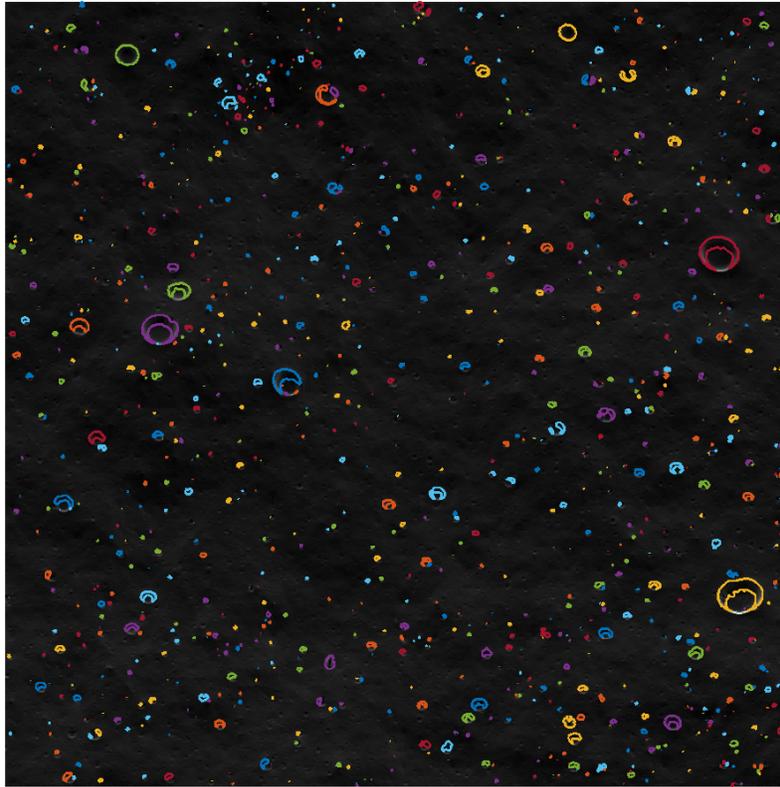


Figure 2.24: The same obstacle detection algorithm was run on this 1-m resolution elevation data from Shackleton crater on the Moon. The algorithm does a much better job in an environment with sharp, clearly-defined obstacles like micro-craters and boulders.

2.5.3 Triangulation

The obstacle detection code writes mapfiles in a particular format, giving the coordinates for the convex hull of each tether obstacle. An operator can then run my tethered motion planning GUI, written in Python, which is pictured in Figure 2.25. The GUI reads the mapfile in as a Doubly-Connected Edge List (DCEL) and displays it, as shown in Figure 2.26.

Next, the tethered motion planner triangulates the map. As described in Section 2.2.2, this involves two steps: dividing the map into y-monotone polygons, as shown in Figure 2.27, and triangulating those polygons. Figure 2.28 shows the resulting BTM. The relatively small scale of the obstacles with respect to the map causes many long, thin BTM triangles that span the width of the map. Despite looking like lines in Figure 2.28, a close-up view of two obstacles in Figure 2.29 reveals that the

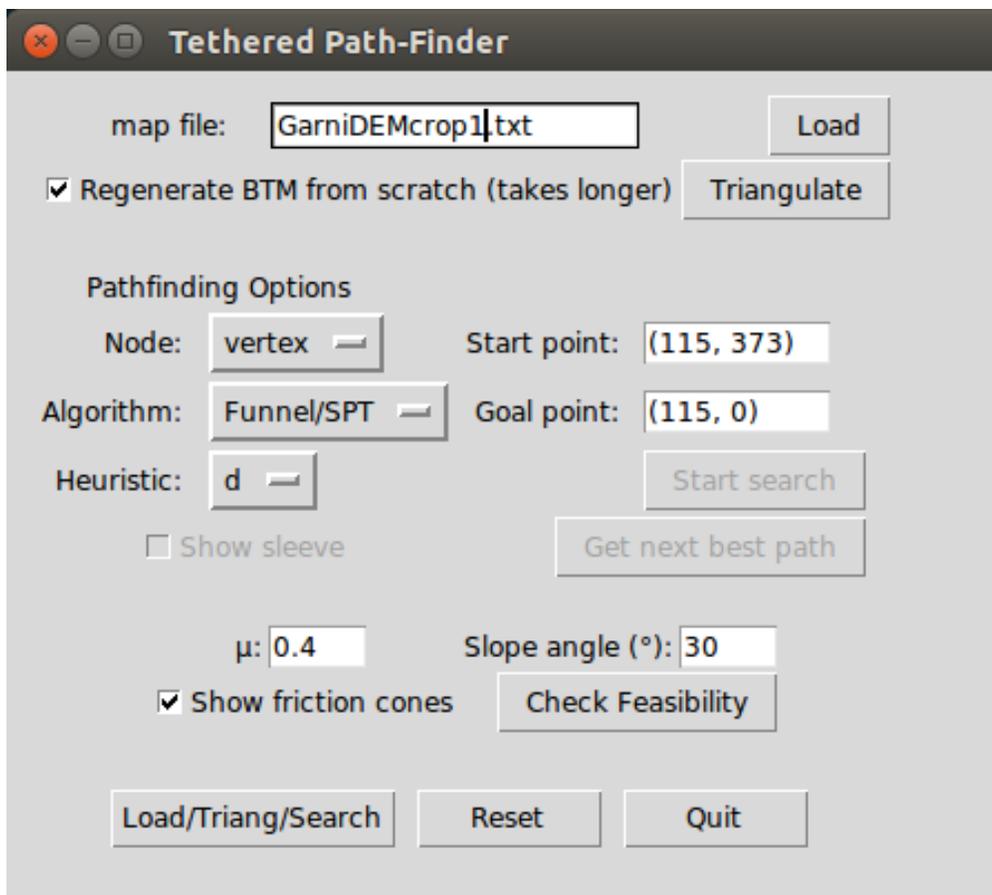


Figure 2.25: The GUI used for tethered motion planning allows an operator to visually check every step of the process, from viewing the map, to triangulating, finding a path, and checking its feasibility. Buttons and checkboxes become enabled after necessary triggering events, to prevent user error.

BTM-finding algorithm has produced triangles, as expected.

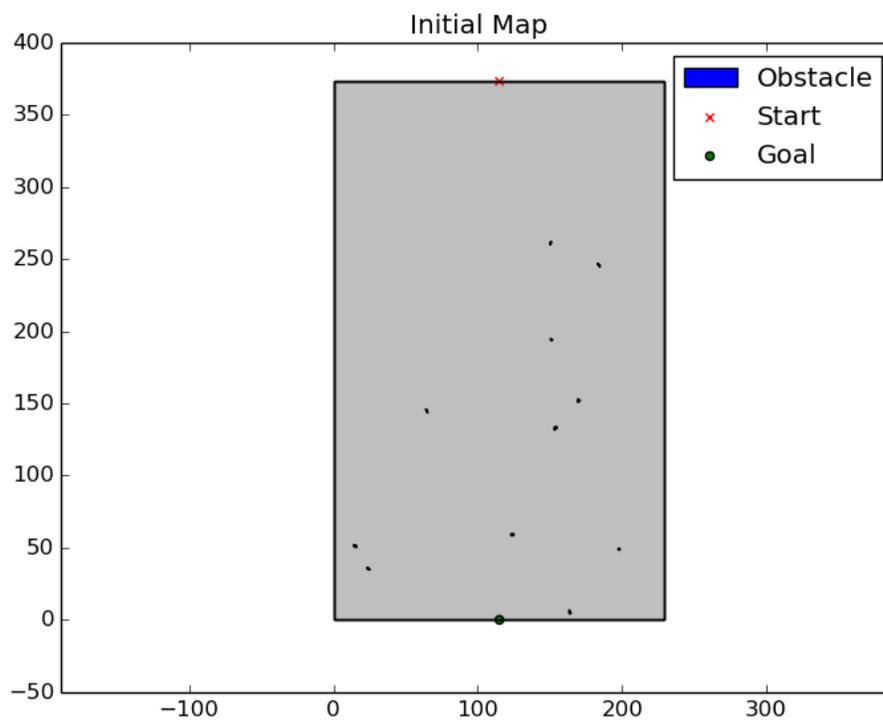


Figure 2.26: The Garni Crater obstacle map resulting from Figure 2.23 is read into the program and displayed. The operator can choose the desired start and end points, depicted by the red 'x' and the green circle, respectively. Any obstacles — tiny on this map, and hard to see — are colored in blue.

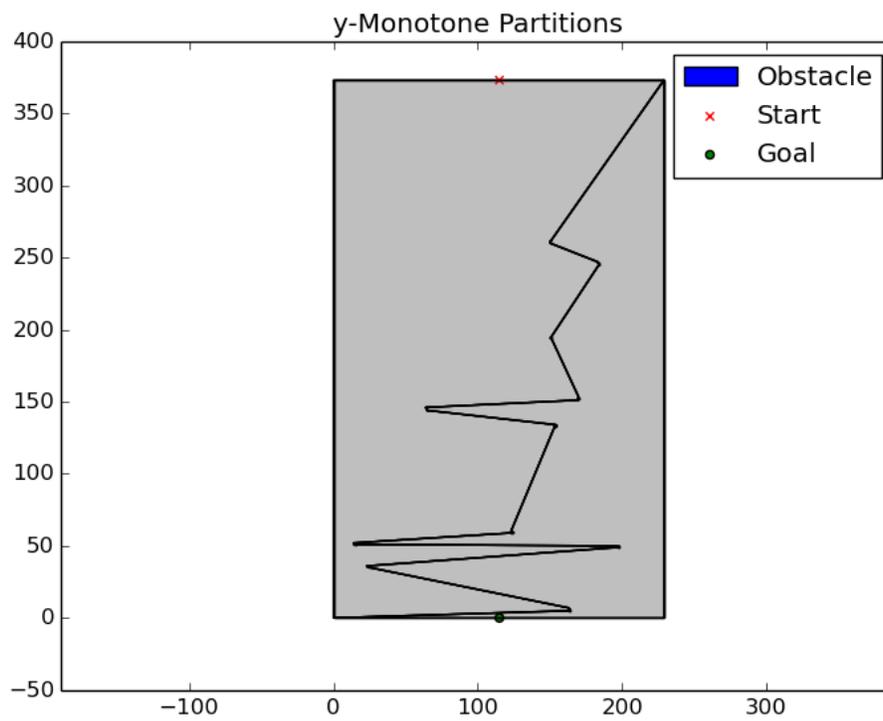


Figure 2.27: The first triangulation step to get the BTM involves dividing the map into y-monotone polygons. This map is useful for debugging, but its display can be turned off for regular use.

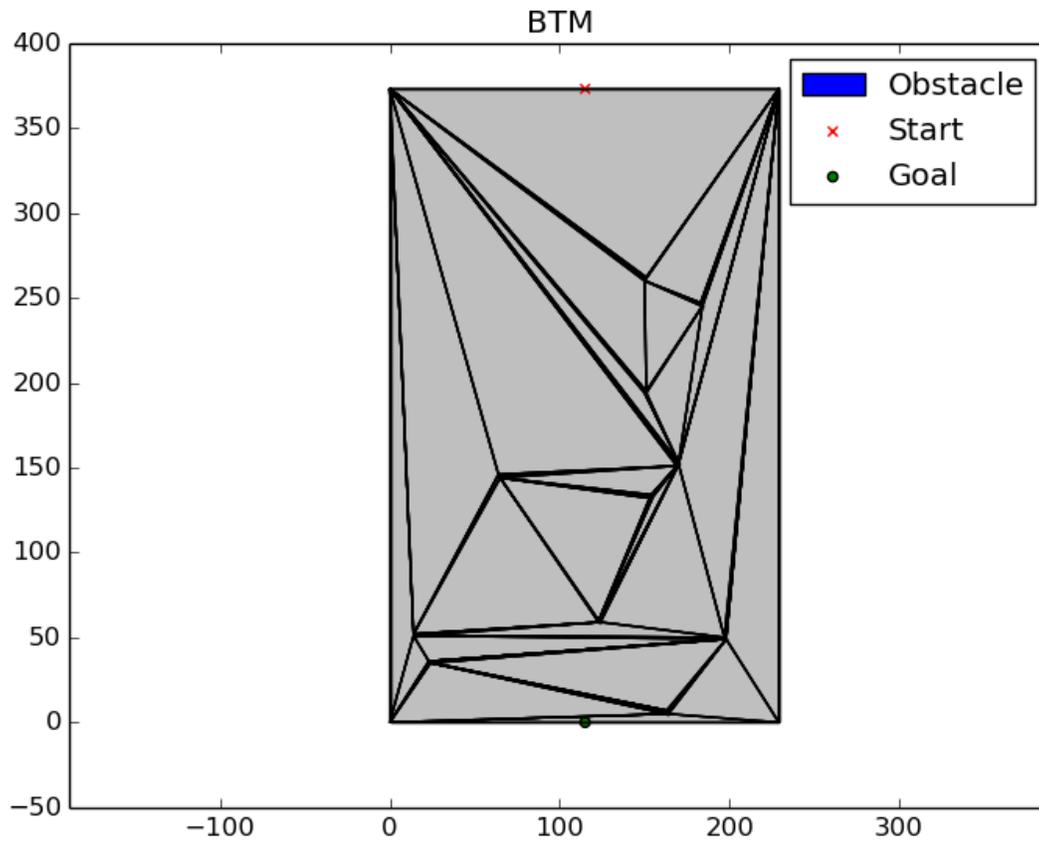


Figure 2.28: Due to the small size of the obstacles, and the large distances between them, many crater BTMs end up looking like this one: with lots of long, skinny triangles near each other.

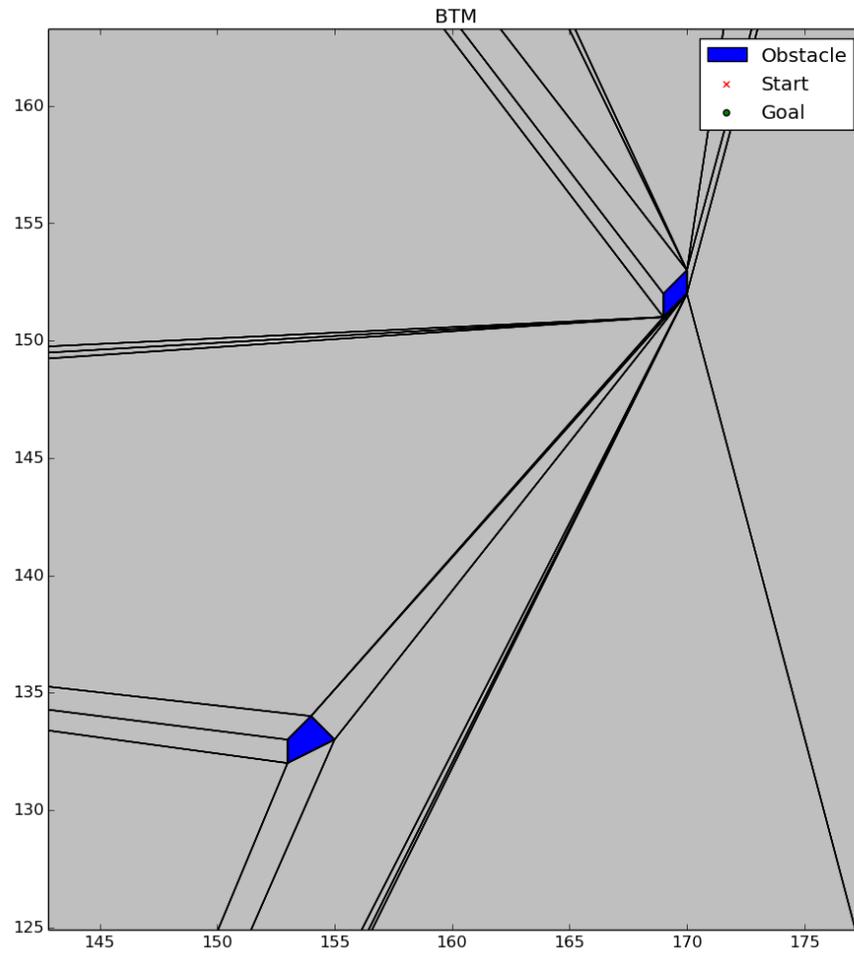


Figure 2.29: A close-up of the BTM in Figure 2.28 reveals that each obstacle does, indeed, have many thin triangles connected to it. Note the scale, in meters, on the side; each of these obstacles is just a few meters wide, for a map whose resolution is 1 m.

2.5.4 Path Search and Feasibility Check

The program then runs an A* search on the shortest path tree. Some examples of the funnels produced in the process can be seen in Figure 2.30, where each new funnel is overlaid on any previous funnels in a different color. As described in Subsection 2.3.3, the program finds the best — in this case, the shortest — SHP. That SHP, shown in magenta in Figure 2.31, is a straight line from the start to the goal, as can be expected in a map with sparse obstacles.

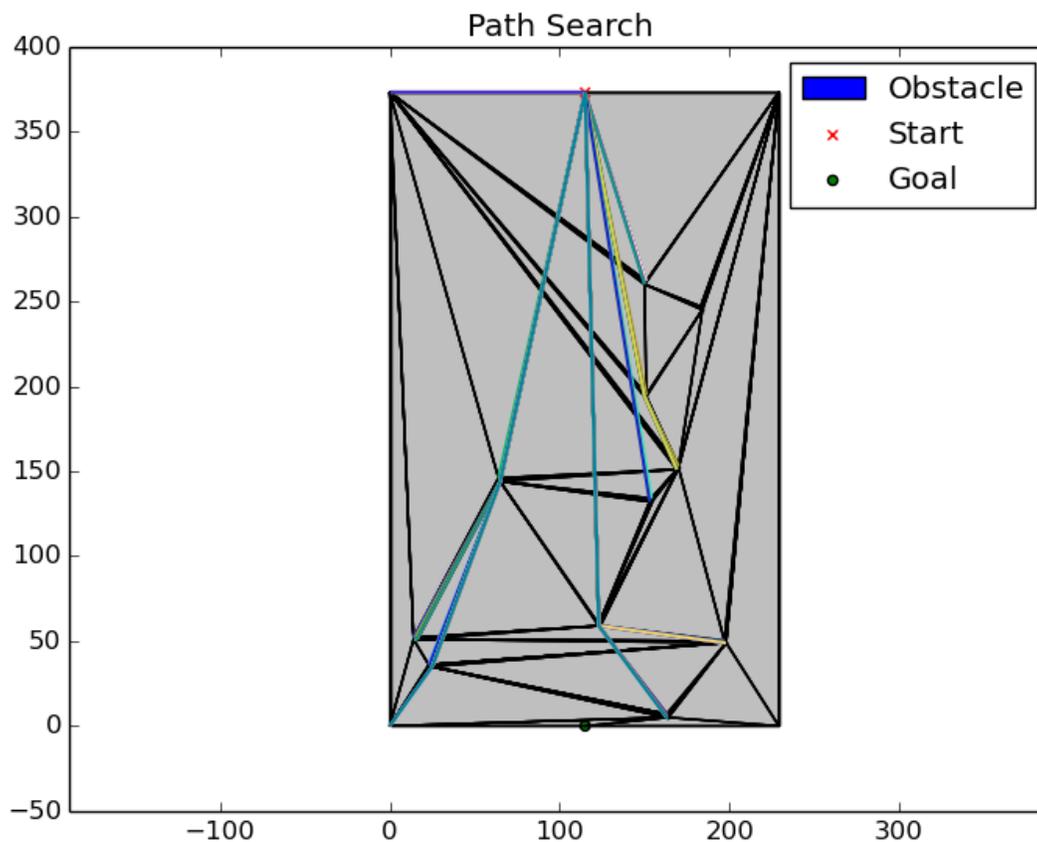


Figure 2.30: This figure is primarily used for debugging the path-finding algorithm. At each step, one of the existing funnels gets overwritten or extended in a new random color. An astute reader may be able to spot some of these funnels-in-progress by their colors in this image.

The GUI in Figure 2.25 gives the operator the option to display the sleeve for a given SHP. Figure 2.32 shows the sleeve for the best SHP, in dark gray. Having found the absolute shortest path from start to goal in the map, the program gives the operator

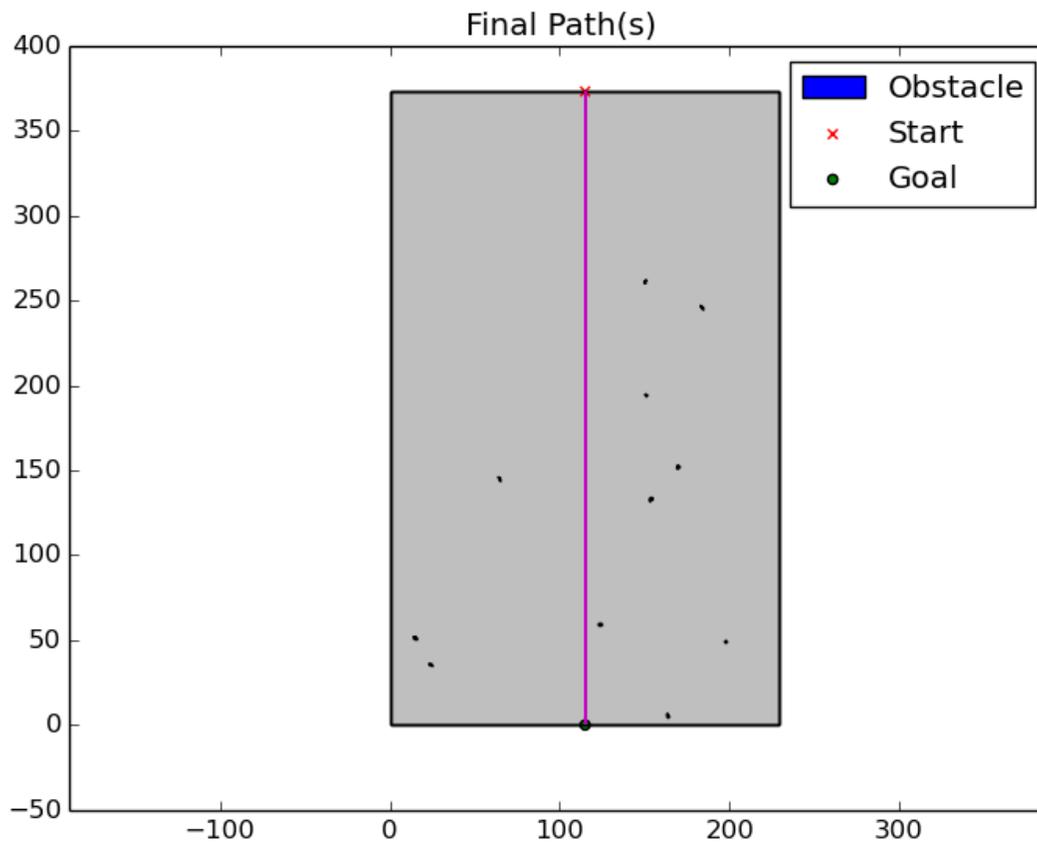


Figure 2.31: Since obstacles were sparse in this Garni map, it is unsurprising that the resulting best (in this case, shortest) path was a straight line from start to goal, as shown in magenta.

an option to continue the search to find the next shortest SHP, or the shortest path that is in a different homotopy class. Figure 2.33 shows the second- and third-best SHPs for this map, along with the sleeve for the third-best SHP.

Due to those slim BTM triangles, a sleeve can often appear from a distance as though it surrounds an obstacle. Closer inspection of the sleeve around an obstacle, as in Figure 2.34 and Figure 2.35, shows that this is not the case; in fact, a thin triangle is excluded from the sleeve.

Once an SHP is found, the program identifies the anchor points and checks if ascent is feasible for that SHP. This requires a slope and friction coefficient, which the operator can enter in the GUI. Figure 2.36 shows resulting wide friction cones

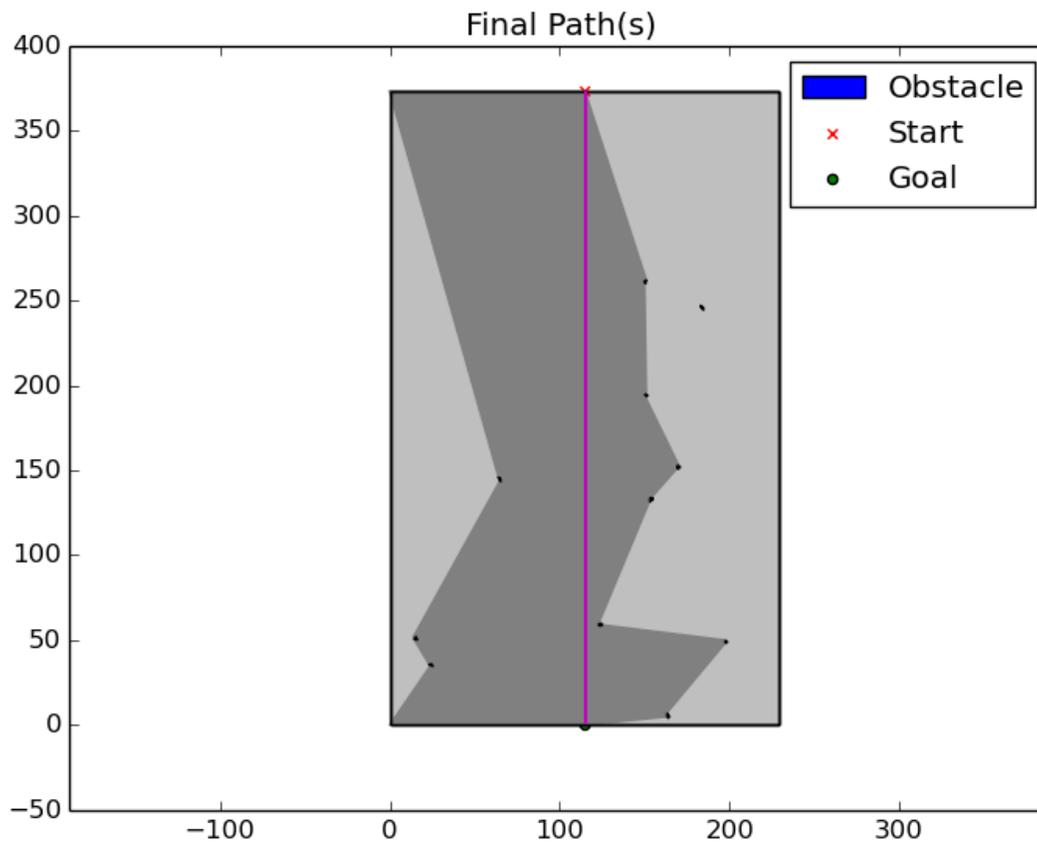


Figure 2.32: The darker gray depicts the sleeve for the best path found in Figure 2.31.

in cyan, for $\mu = 0.4$ and a 45° slope. Figure 2.37 is the same situation, except with $\mu = 0.3$. Intermediate anchors are shown as yellow squares, and at least one waypoint along the visibility line is given for each intermediate anchor. Although their similarity in size to the obstacles might make them hard to see, both figures depict waypoints as stars.

Since the resolution of these satellite images is too low for precise robot planning, the tethered motion planner acts as a high-level task planner. The sleeves and waypoints can then be passed to a motion planning algorithm that incorporates higher resolution sensory data from Axel and uses it to plan paths between waypoints.

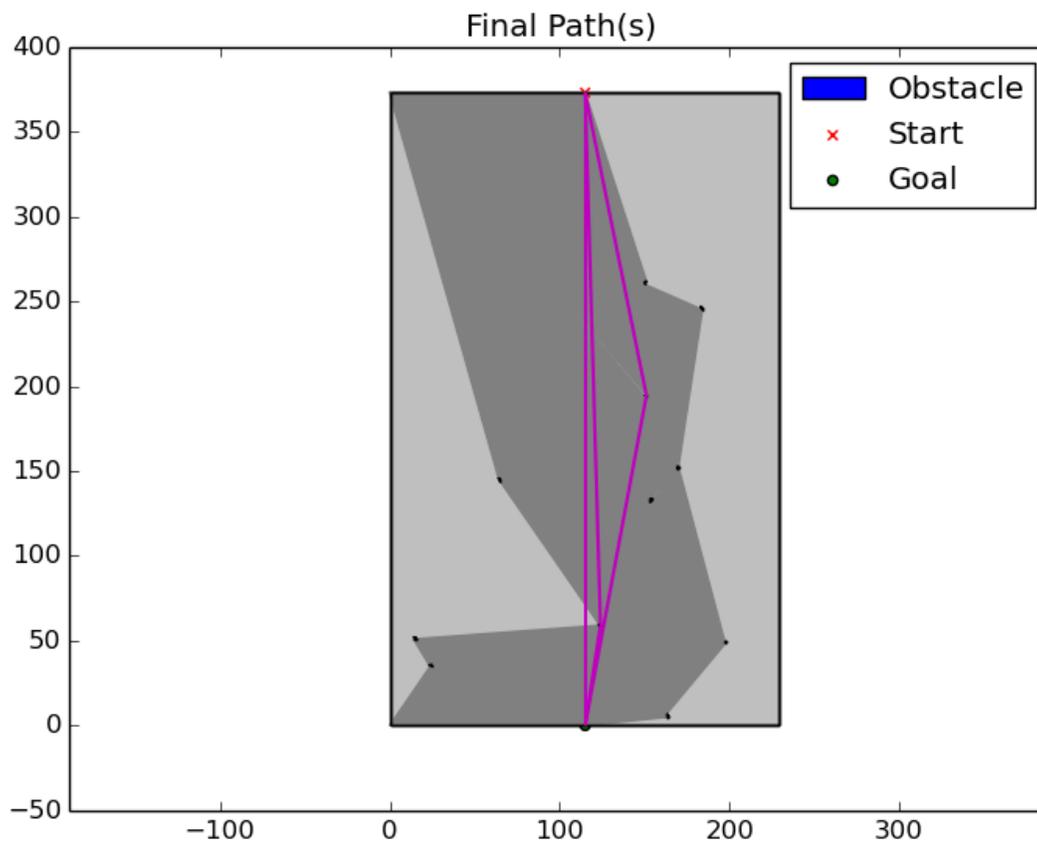


Figure 2.33: The best, second best, and third best paths are shown in maroon. In dark gray is the sleeve for the third best path.

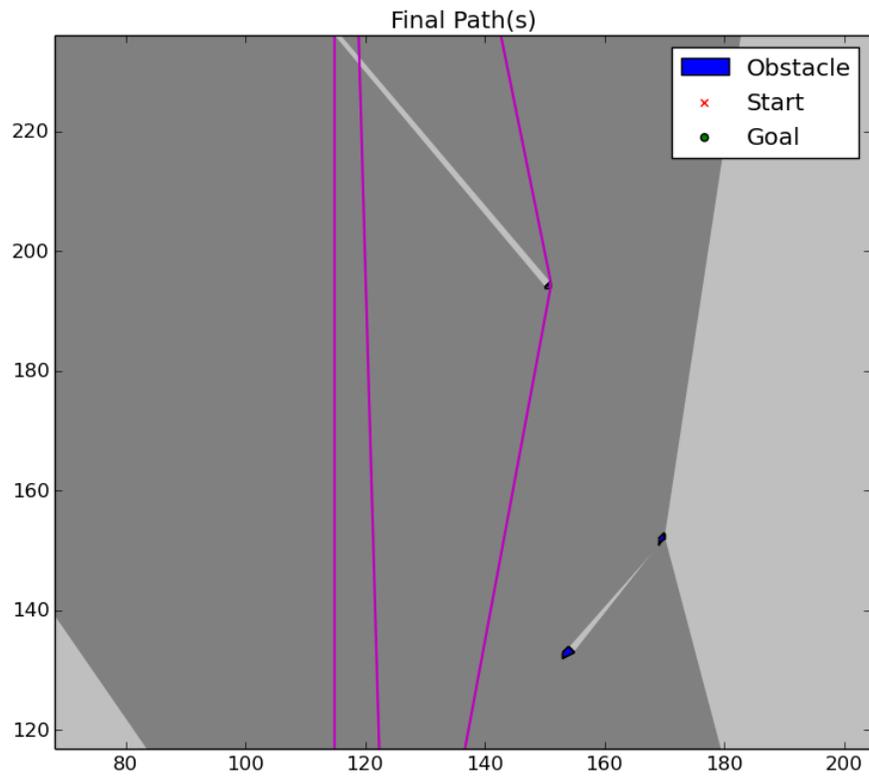


Figure 2.34: Although the sleeve in the previous image may appear to encompass an obstacle, a closer look reveals that there is actually a very small, slim BTM triangle that is not part of the sleeve.

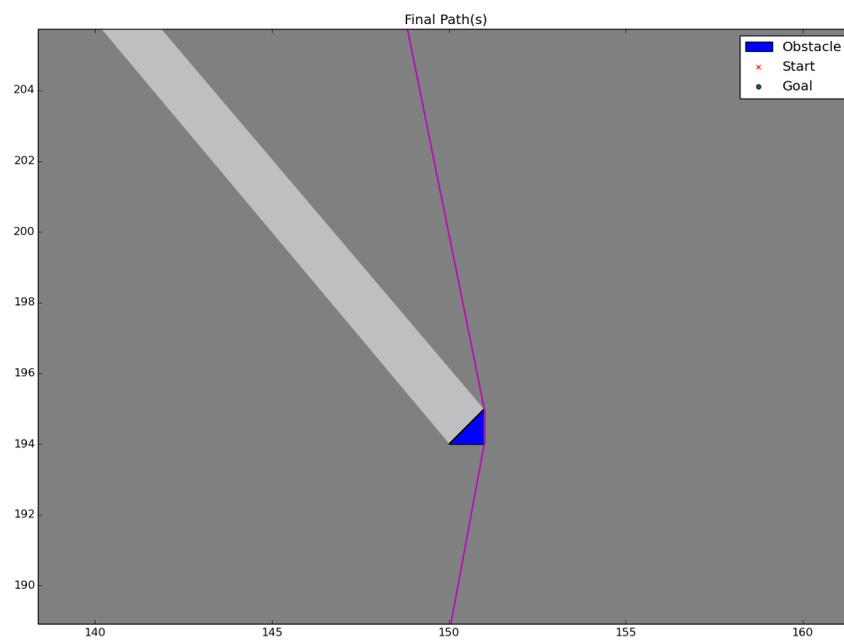


Figure 2.35: An even-closer look at the obstacle in question.

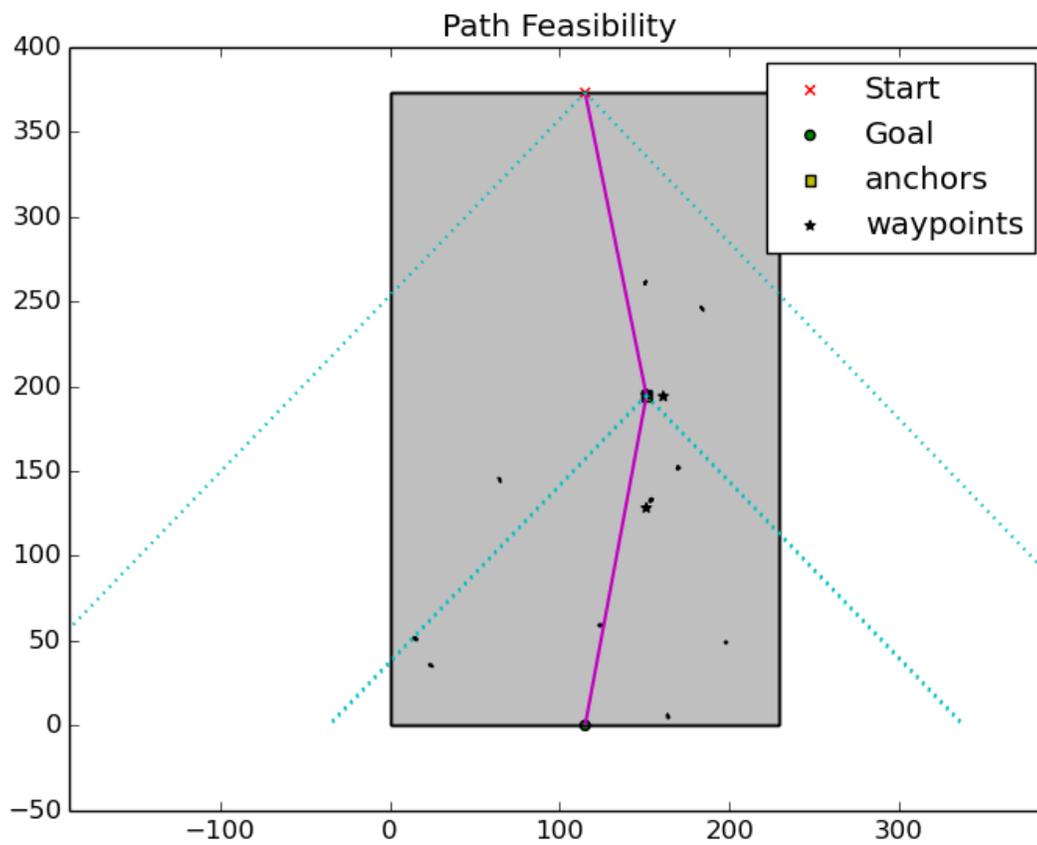


Figure 2.36: This path is feasible; it easily falls inside the wide cyan lines, resulting from the $\mu = 0.4$ on a 45° slope.

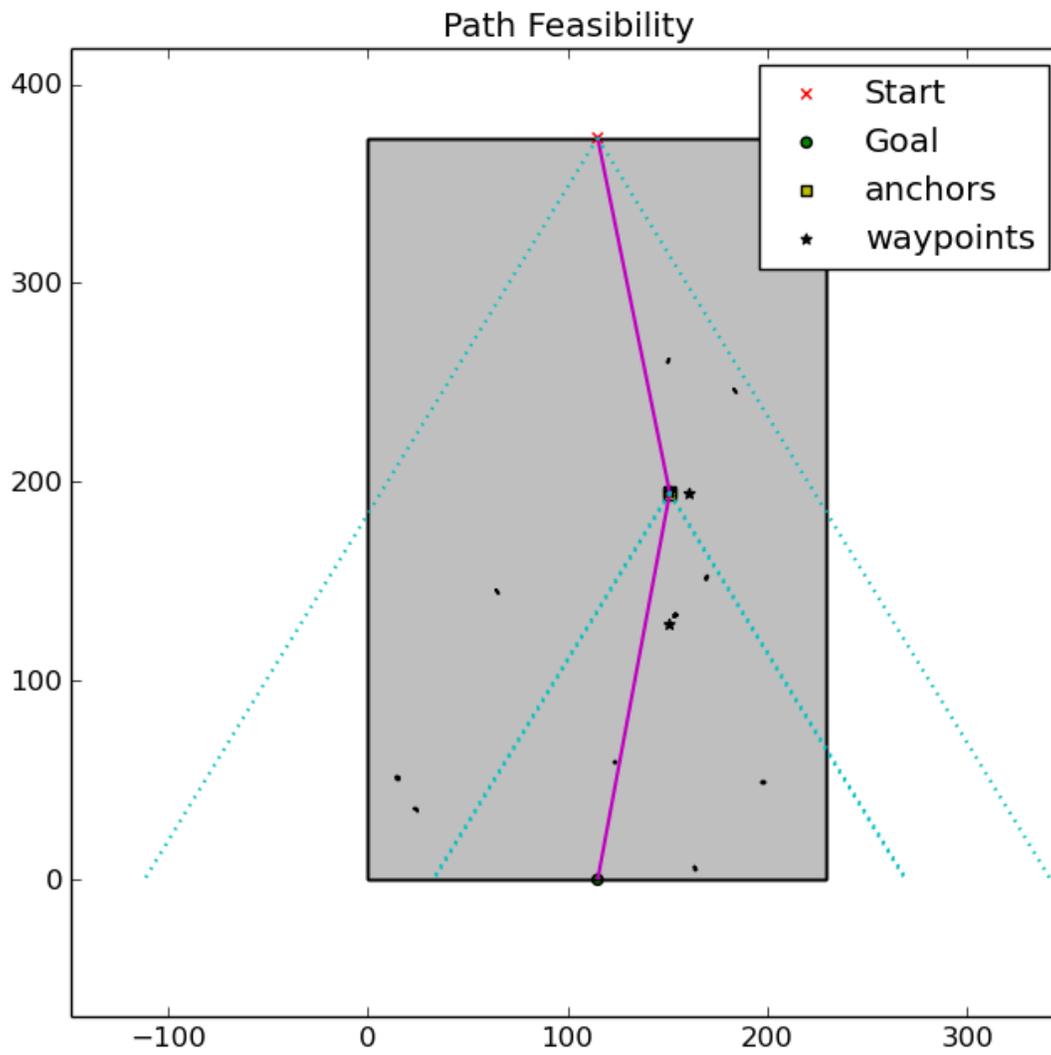


Figure 2.37: Even with a smaller friction coefficient, this path is feasible too.

2.5.5 Golden Queen Mine

To better anticipate potential issues on an extraterrestrial mission, Axel has been strenuously field-tested on Earth. At the time of this analysis, the latest field test has been scheduled for August of 2019; the team plans to pilot the robot down the steep, rocky slopes of a Meteorology Station Hill at Golden Queen Mine, in the California desert north of Lancaster. In this analysis, the aforementioned program was used to prove that safe, feasible paths exist at that site. In addition to its eventual use for full autonomy, this software could be used on site as an additional planning tool for human operators during any future field tests.

Figure 2.38 shows a DEM of the testing site at Golden Queen Mine. As in Subsection 2.5.2, a Matlab obstacle detection program was run on a cropped subset of the DEM. Figure 2.39 shows the resulting obstacles which form the starting map for path planning.

Figure 2.40 shows the BTM for the Golden Queen Map. For clarity, a zoomed-in selection of the top of the map is shown in Figure 2.41. Similarly, Figures 2.42 and 2.43 show the shortest path from the start to goal, and the sleeve for that path; the latter is a close-up of the topmost portion of the map. The shortest path is magenta, and the sleeve for that path is dark gray.

Despite its first appearance, this final shortest path is not a perfect straight line. The 2 small obstacles that form anchors can most clearly be seen in Figure 2.44, as gold boxes. That figure also depicts the wide friction cones with $\mu = .4$, meaning that this path is feasible.

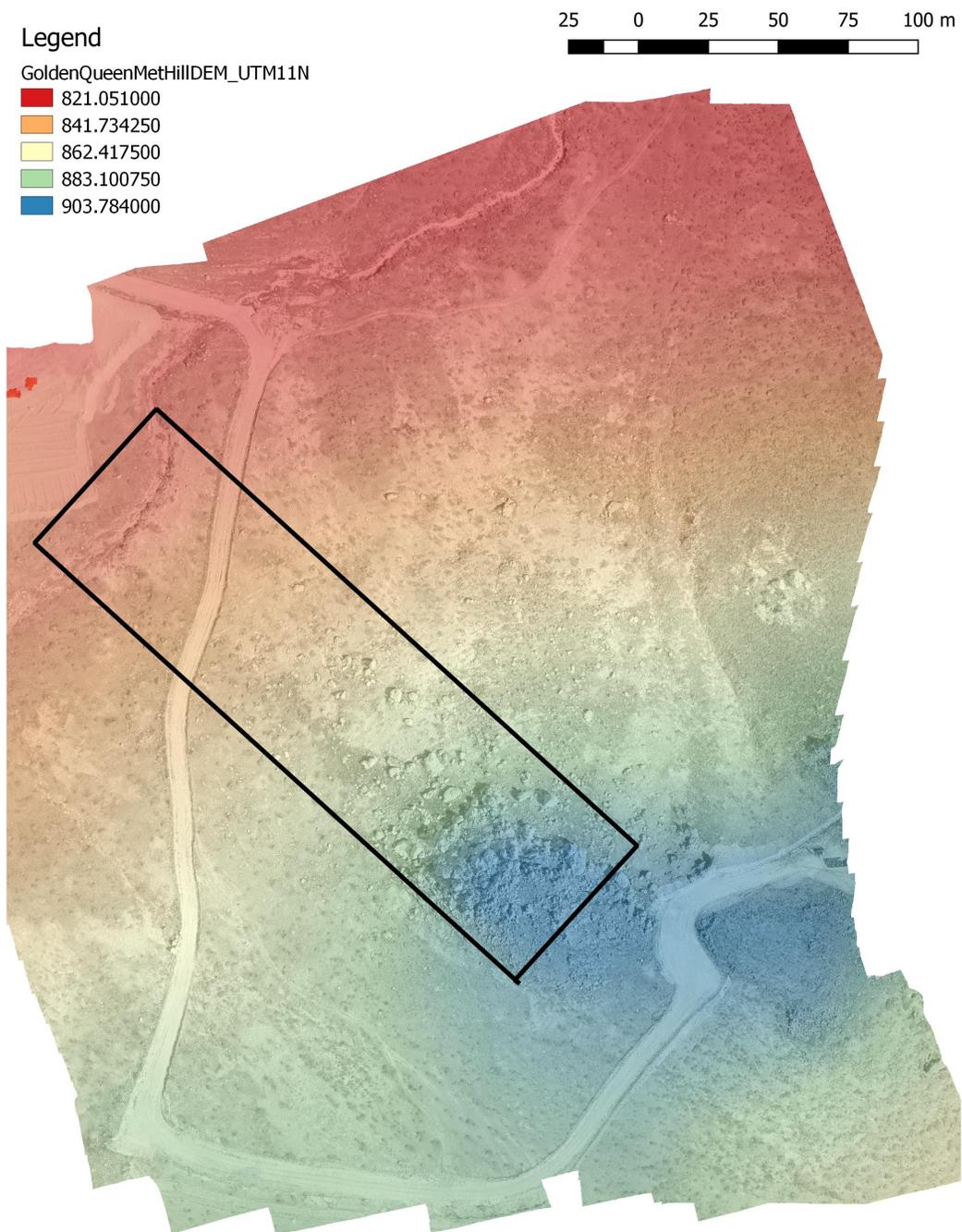


Figure 2.38: This DEM of the testing site at Golden Queen Mine is based on data from the Harvard Dataverse, by Sparky Henderson. The black rectangle represents the cropped area that was used as the robot's map.

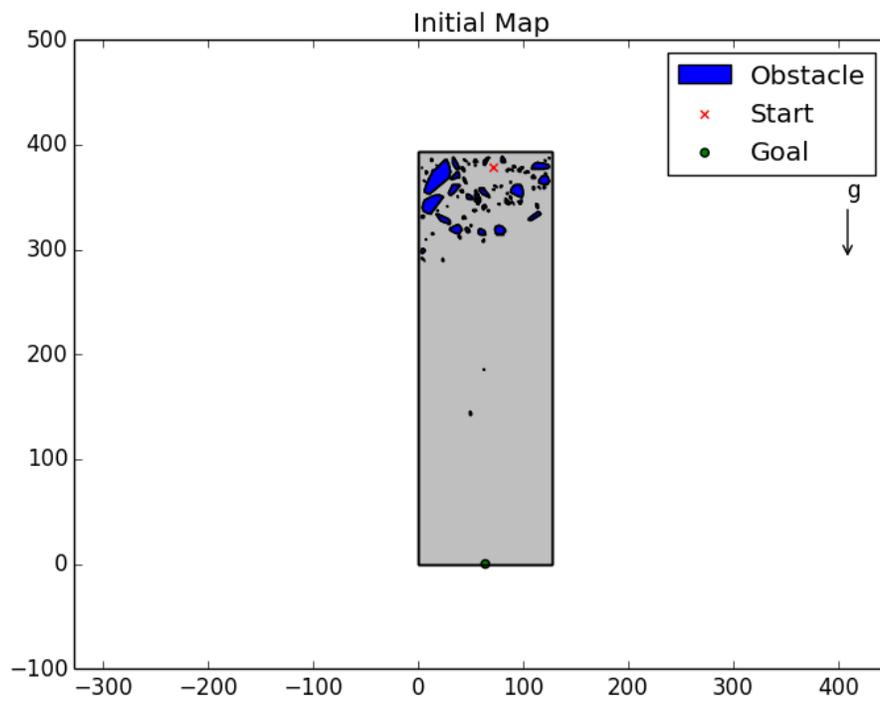


Figure 2.39: The Golden Queen Mine obstacle map, as displayed by the path-planning program. As in Subsection 2.5.3, obstacles are shown in blue. An example starting point is denoted with a red 'x', and a desired goal point is denoted by the green circle.

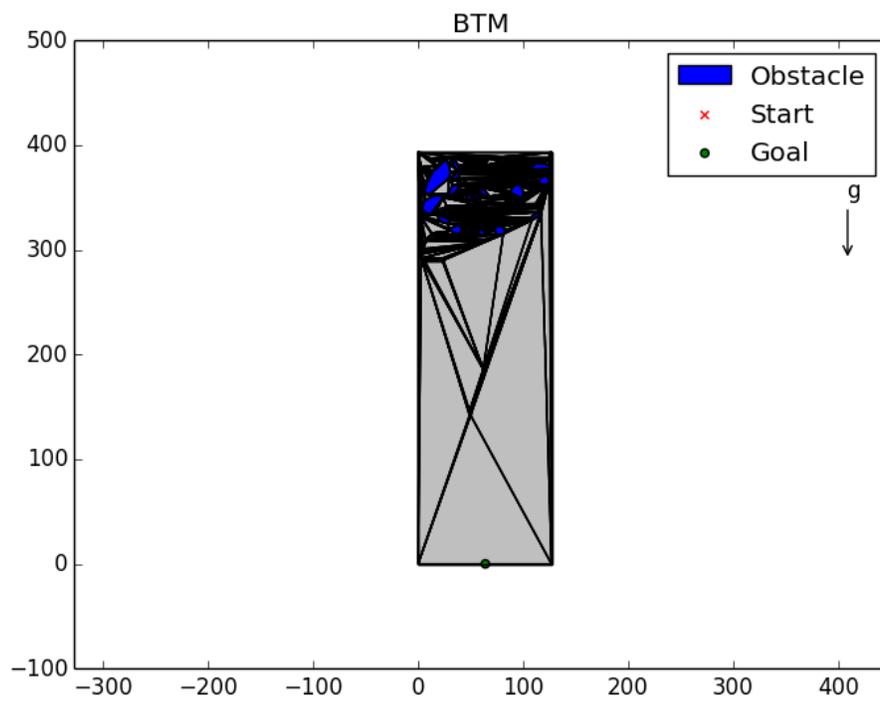


Figure 2.40: The BTM for the Golden Queen Mine map.

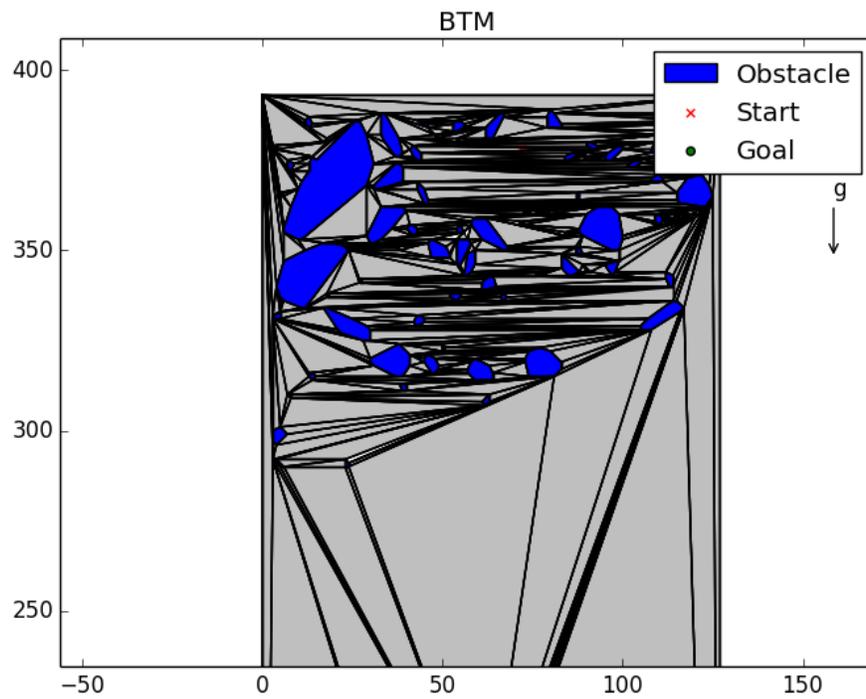


Figure 2.41: Zooming in on a section of the BTM gives a better sense of the triangle structures that compose it.

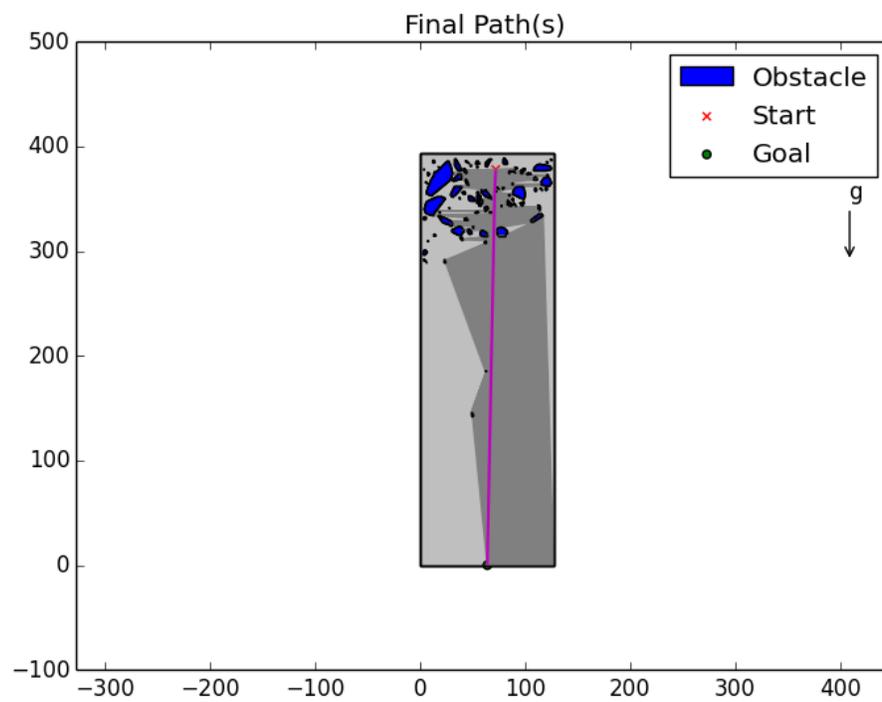


Figure 2.42: Although it appears that the path between start and goal is a straight line, there are in fact 2 small obstacles around which the robot must detour. This can more clearly be seen in the close-up views, and in the fact that Figure 2.44 shows 2 anchors.

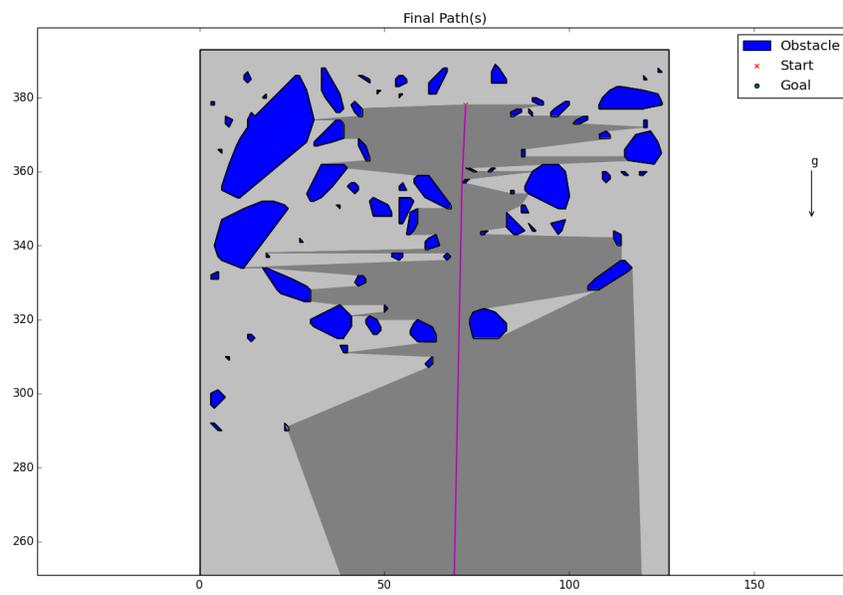


Figure 2.43: A close-up view of the final path and its sleeve.

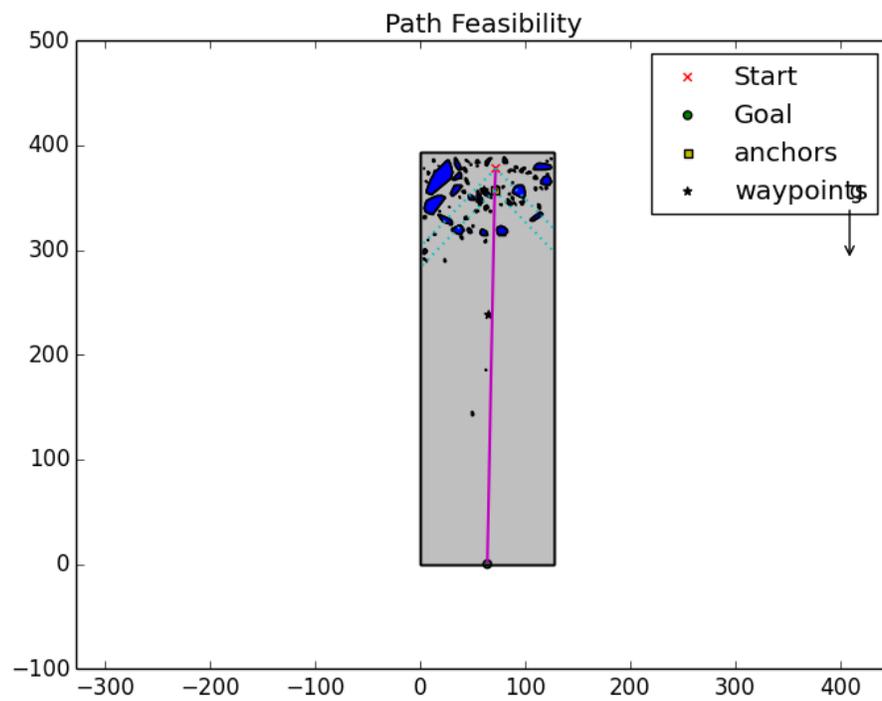


Figure 2.44: The friction cones, in cyan, extend from the 2 anchors boxed in gold. Since each anchor is within the friction cone of its proximal anchor, the robot can safely ascend this path.

2.6 Summary

This chapter explained one of the key concepts driving tethered motion planning: to avoid leaving the tether tangled around an obstacle, the planned ascent and descent paths must be homotopic to each other. Those homotopy classes can be represented by their shortest path, the SHP, or by their sleeve (among other things). The Funnel Algorithm can be used to find the shortest path in a sleeve, and the Guibas' Shortest Path Tree algorithm can find the shortest paths to all the points in a polygon. Hershberger and Snoeyink showed that, using a BTM, these algorithms can be extended to find all the shortest paths in a polygon with holes.

My contribution was to apply graph-search techniques to these algorithms, adding a queue and a heuristic. By treating certain BTM vertices as nodes, the SPT algorithm can be altered to return the shortest SHP, followed by the next shortest, and so on. Each of these SHPs in turn represents a homotopy class, so this search can be used to quickly narrow down the sleeve in which the robot should search for ascent and descent paths. That SHP also forms the basis for the feasibility check, in which the algorithm must check each anchor point to ensure that it is passable. I've presented the conditions for which this is true, with a simplified friction cone model of the area the robot can traverse. In the process of checking if ascent is feasible, the algorithm can easily return a set of waypoints that are on the lines of visibility for each anchor. A more precise motion planner, with more detailed data, can then find a pair of paths in the sleeve that pass through those waypoints.

This process even works for a series of planes, as long as the edges between those planes are convex, such that the tether stays on the ground. Section 2.5 shows the intermediate steps and the paths found for some actual Martian terrain, using the program I developed for that purpose. Overall, this chapter shows how to preprocess DEM data, find a BTM, and create and search a Shortest Path Tree to find an SHP. After that SHP is checked for feasibility, a sleeve and a set of waypoints can be found, in the process of finding or generating an ascent and descent path pair.

Chapter 3

MODELING AND PREDICTING TETHER-TERRAIN INTERACTION

3.1 Introduction

Chapter 2 presented a tethered motion planning algorithm under the assumption of a frictionless tether. In practice, however, the friction between the terrain and the tether can have a significant impact on the behavior of the rope, and therefore on the robot's movement. As Axel descends the slope, the tether can catch on raised patches of the terrain. These *tether friction obstacles*, described in Subsection 3.1.1 below, differ from the obstacles in the previous chapter because they only occur in certain tether configurations with a sufficiently high friction coefficient between the tether and the terrain.

Tether friction obstacles change the tethered motion planning problem in several ways. If a planner fails to account for them in choosing a descent and ascent path, those paths may end up on different sides of that obstacle. That new obstacle would, in effect, split one homotopy class (and one sleeve) in two, and Axel could not take its planned ascent path without wrapping the tether around the obstacle. Even if this friction obstacle occurred in a location such that the ascent and descent paths were still homotopic, it could cause other problems. That obstacle might create new anchor points, which necessitate reevaluating the passability of those anchor points and the feasibility of the SHP. These, in turn, could change the waypoints on the planned ascent path.

Furthermore, tether friction obstacles pose a potential danger to a rover. Because tether friction depends on tether position and tension, Axel's movements may jostle the tether over or past a friction obstacle, causing the robot to suddenly slip downwards. Thus, any motion plan should either avoid tether friction obstacles entirely, or ensure that those obstacles will remain obstacles (or non-obstacles) in the face of perturbations in tether tension and position.

In this chapter I introduce the concept of tether friction obstacles, and give a general overview for how to plan with tether friction. Next I review and organize the scant existing literature on rope physics, which give equations for the frictional forces on the tether as a function of tether shape and tether tension. With a series of

computational examples, I show how these equations can be used to determine the limits of tether tension for which there is static equilibrium or, given tether tension, whether the tether is in static equilibrium. The latter forms the basis for any frictional tethered planning algorithm, in which potential tether friction obstacles must be evaluated to determine if the tether will stick. After describing the necessary assumptions made about tether tension, I give pseudocode in Section 3.4 for path planning with a frictional tether. To the planner, tether friction obstacles may seem to appear or disappear after the map has been triangulated, so Section 3.5 explains how to update BTMs and sleeves to handle such changes. I also give an algorithm for online tethered path planning that is based on the same concepts, taken from [53]. Finally, I conclude with a summary and discussion of the challenges involved in planning with a frictional tether.

3.1.1 Tether Friction Obstacles

The geometric tether obstacles analyzed in Chapter 2 acted as permanent obstacles to the tether due to their shape and height, and could be represented by convex polygons or as the union of convex polygons. Because Axel’s caster arm can raise the height of the tether to a level higher than its wheels, it was assumed that any tether obstacle was also a robot obstacle.¹ However, there is another category of tether obstacles: *potential tether friction obstacles* (PTFOs). These PTFOs are places where the terrain may, in the right circumstances, act as an obstacle to the tether, but not necessarily to the robot. In this case, the presence of the PTFO prevents the tether from sliding along the terrain surface in the regions of the PTFO, thereby acting as both an obstacle to tether movement and as a potential anchor point for the rover.

For example, consider a tether under tension looped around a low bump or hummock. With no friction between the tether and the hill, the tether will slide right over the top. In this case, that hillock does not act as an obstacle. With sufficiently high tension, the tether will do the same; at infinite tension, the tether converges on the same configuration as in the frictionless case.² However, at low tensions with a high enough friction coefficient, the tether may catch on the hill. Whether or not that bump acts as an obstacle depends on the friction coefficient, the tether’s starting

¹Due to its grousers, Axel can sometimes climb obstacles even taller than its wheels, if that obstacle’s surface provides “footholds” for said grousers. Similarly, the arm generally pushes against the ground during movement, providing an extra lever arm. This definition of a tether obstacle assumes that Axel can lift its arm to move its tether over an obstacle if needed.

²further discussed in Subsection 3.2.5.

configuration, the terrain’s local geometry, and the tension (and change in tension) along the tether.

One of the contributions of this chapter is to analyze how a given tether interacts with a PTFO to determine if that PTFO does, in fact, act as an obstacle under those circumstances. If it does, it is known as a *tether friction obstacle* (TFO), and the points at which the SHP contacts that obstacle as *friction anchors*. These TFOs should still be treated differently from geometric obstacles for several reasons. Because these obstacles depend on friction, they only occur for a tether lying on the ground. In practice, this doesn’t always happen. As Axel drives, its caster arm frequently moves up and down to go over rocks or to adjust tension. This means that anchors around these obstacles are easily lost, especially while the robot is near that obstacle. In addition, these tether obstacles generally are not robot obstacles. This means that, rather than traveling around an anchor to remove it, Axel can drive right over that anchor point.

PTFOs can be helpful or harmful in planning and execution. Obviously, if Axel unexpectedly loses an anchor at a PTFO, the tether may suddenly go slack. If this occurs on a tether-demand plane, the robot may slip downhill, perhaps into an undesirably dangerous area. Conversely, sometimes the operator may wish for an obstacle when there is none. For example, a relatively steep slope means relatively thin friction cones, limiting Axel’s range from side to side. However, if it is preceded by a gentler slope, anchoring on an obstacle to one side of that slope may allow Axel to reach areas on the steep slope that it otherwise could not. If no traditional obstacle is present, a friction obstacle may provide this opportunity. Indeed, this is when friction obstacles work best; the additional contact angle as the tether goes over the cliff ensures that the friction obstacle will experience less tension as Axel moves past that cliff edge, which acts like a capstan.³

3.1.2 Planning with Tether Friction

In preprocessing a map without tether friction, the first step is tether obstacle detection. When incorporating tether friction, this step should also identify PTFOs, and label them as such. This simply means finding any convex terrain, by analyzing the surface curvature as discussed in Subsection 3.3.3. In both cases the map, represented by a polygon with holes P , is next triangulated into a BTM, B .

The algorithm described in Section 3.4 first searches for any PTFOs. It will treat

³For more information on capstans, see Subsection 3.2.5.

them like geometric tether obstacles as it finds an SHP. Then, it checks how the tether configuration represented by that SHP will interact with any PTFOs. Section 3.2 describes the rope physics equations needed to determine if the tether will stick or slip in those cases. Section 3.3 shows some simple examples of ropes under tension, and computes the tension changes at static equilibrium or whether those ropes will slip. That check also requires some assumptions about tether tension, which are described in Subsection 3.4.1.

If the tether slips off an anchor point, the SHP must be replanned around it. This does not mean that the planner can simply remove that anchor point from the list. The taut tether may actually contact another obstacle that was occluded by the first. This means that updating an SHP to remove a PTFO requires several elements from online motion planning, like re-triangulating an area on the fly, and finding a new sleeve. These are described in Section 3.5.

If the SHP will stick, that PTFO becomes a TFO. Each anchor point can then be checked to see if it is passable, and the whole ascent feasible. This involves another step for TFOs, however; because sticking is dependent on tether configuration, the planner must check for TFO stability at every angle the tether passes through as Axel moves. Finally, if the SHP passes all these tests, the planner can generate waypoints and return the sleeve, just like in Chapter 2. This process is described in further detail in Section 3.4.

Clearly, every element of planning with a frictional tether requires a solid understanding of how ropes interact with frictional terrain, so let us begin with a literature review on the subject.

3.2 Physics of Rope on a Smooth Frictional Terrain

Given a tether position, tension, and friction coefficient, what are the frictional forces on the tether? Will that tether stay in static equilibrium? These questions are answered by the force balance equations for a rope with a constant friction coefficient described by Maddocks and Keller [73]. Konykhov used variational calculus to derive the same equations for orthotropic surfaces, in which the surface has a direction-dependent friction coefficient [74]. He described the requirements for a rope to stick, and gave the dynamical equations of motion for a sliding rope.

This section summarizes relevant parts of their work. It first discusses the frames of reference commonly used for a curve on a surface, which are what the aforementioned authors used in formulating the statics equations of a rope. The next

subsection explains some of the necessary assumptions made about the frictional forces and the rope model. Subsection 3.2.3 follows Maddocks' and Keller's derivation of the force equations for a rope in static equilibrium. These equations can be greatly simplified by assuming that the rope is weightless, which results in a single simple equation for static equilibrium. As will be seen in Section 3.3, that equation can be used to determine if a tether in a given configuration will stick or slip, and therefore whether a certain PTFO acts as a tether friction obstacle. Finally, the last subsection describes a few special cases relevant to the derived force-balance equations, to provide insight on some of their implications for tethered planning.

3.2.1 Frames of Reference

This subsection reviews some useful frames of reference for describing tether physics. For more details on the differential geometry of curves and surface, see [98].

Differential geometry commonly uses the Frenet-Serret frame to describe the motion of a particle along a continuous differentiable 3-D curve, as in Figure 3.1. This frame is composed of the unit vectors $\hat{\mathbf{T}}$, $\hat{\mathbf{N}}$, $\hat{\mathbf{B}}$, or the tangent, normal, and binormal unit vectors, respectively. For a curve $\rho(s)$ that is arc-length parameterized⁴ by s , those basis vectors can be defined as

$$\begin{aligned}\hat{\mathbf{T}} &\triangleq \frac{d\rho}{ds} \\ \hat{\mathbf{N}} &\triangleq \frac{1}{\kappa} \frac{d\hat{\mathbf{T}}}{ds} \\ \hat{\mathbf{B}} &\triangleq \hat{\mathbf{T}} \times \hat{\mathbf{N}}.\end{aligned}\tag{3.1}$$

At every point along that curve, its curvature, κ , is defined as

$$\kappa(s) \triangleq \left\| \frac{d\hat{\mathbf{T}}}{ds} \right\| = \|\rho''(s)\|.\tag{3.2}$$

These 3 vectors form a frame that moves along the curve as s increases. At any given point s_0 , the vectors $\hat{\mathbf{T}}$ and $\hat{\mathbf{N}}$ span a plane containing s_0 , called the *osculating plane*. The *osculating circle*⁵ is a circle through s_0 in this plane, with its center in the direction of $\hat{\mathbf{N}}$. Its radius R is $\frac{1}{\kappa(s)}$.

⁴An arc-length parameterized curve has unit speed: $\|\rho'(s)\| = 1$.

⁵The Latin etymological root *osculum* means “to kiss”; in this case, the circle “kisses” the curve $\alpha(s)$ at s_0 .

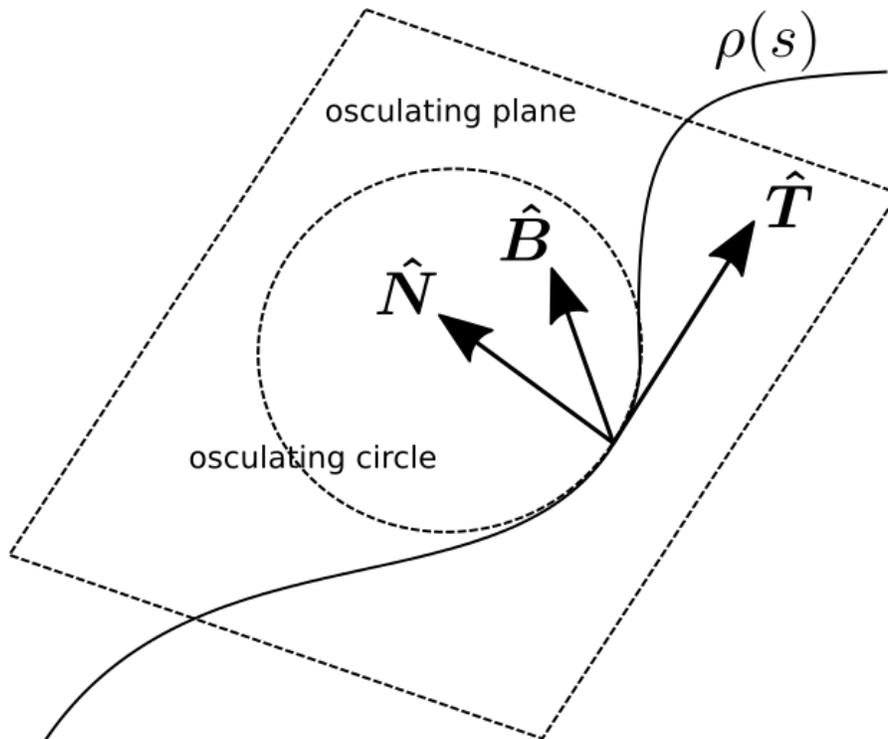


Figure 3.1: The Frenet-Serret frame for a curve $\rho(s)$ in 3-D space.

Now assume that, rather than moving in free space, the arc-length parameterized curve $\rho(s)$ lies on some 3-D surface M , like in Figure 3.2. Like the Frenet-Serret frame, the Darboux frame follows the motion of that curve. Unlike the Frenet-Serret frame, it incorporates the surface geometry, with basis vectors that are oriented tangent to the curve and normal to the surface.

$$\begin{aligned}
 \hat{\mathbf{t}} &\triangleq \frac{d\rho}{ds} \\
 \hat{\mathbf{n}} &\triangleq \nabla M \\
 \hat{\mathbf{g}} &\triangleq \hat{\mathbf{t}} \times \hat{\mathbf{n}}
 \end{aligned}
 \tag{3.3}$$

Note that the Frenet-Serret and the Darboux tangent unit vectors are identical: $\hat{\mathbf{T}} = \hat{\mathbf{t}}$. The Darboux normal is the surface normal, *not* the curve's normal: $\hat{\mathbf{N}} \neq \hat{\mathbf{n}}$. In fact, the two normal vectors differ by some angle ϕ around $\hat{\mathbf{t}}$. The $\hat{\mathbf{g}}$ unit vector — called the tangent normal, or the geodesic unit vector — is orthogonal to both the tangent and the normal unit vectors.

The normal curvature, κ_n , is the curvature of $\rho(s)$ in the $\hat{\mathbf{n}}$ direction. The geodesic curvature, κ_g , is its curvature in the $\hat{\mathbf{g}}$ direction. In other words, the normal and

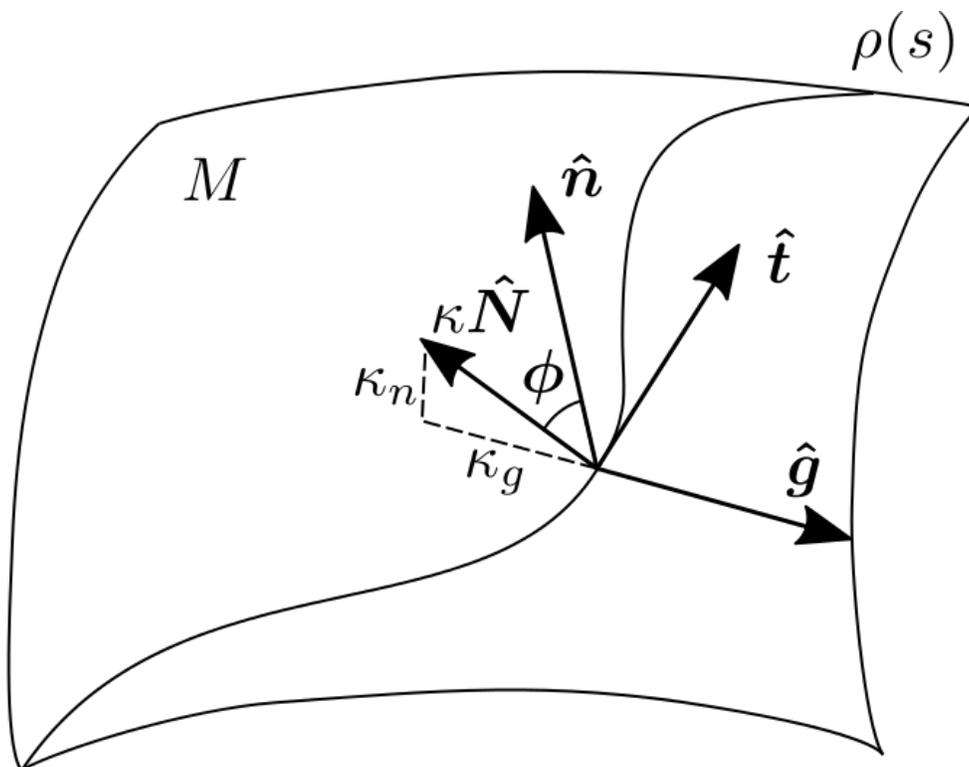


Figure 3.2: The Darboux frame for a curve $\rho(s)$ on a surface M .

geodesic curvatures are the components of $\kappa \cdot \hat{\mathbf{N}}$ projected onto the surface M .

$$\begin{aligned}\kappa_n &= \frac{d\hat{\mathbf{t}}}{ds} \cdot \hat{\mathbf{n}} = \rho''(s) \cdot \hat{\mathbf{n}}, \\ \kappa_g &= \frac{d\hat{\mathbf{t}}}{ds} \cdot \hat{\mathbf{g}} = \rho''(s) \cdot \hat{\mathbf{g}}.\end{aligned}\tag{3.4}$$

The previous two equations can also be written in terms of ϕ , the angle between $\hat{\mathbf{N}}$ and $\hat{\mathbf{n}}$:

$$\begin{aligned}\kappa_n &= \kappa \cos(\phi), \\ \kappa_g &= -\kappa \sin(\phi).\end{aligned}\tag{3.5}$$

Note that a curve is a geodesic if $\kappa_g = 0$, and therefore $\kappa = \kappa_n$. A geodesic curve's acceleration is always normal to the surface. Therefore κ_g can be thought of as describing how far a curve is from being a geodesic.

3.2.2 Rope Model

Model a rope or tether lying on the ground as a long, thin cylinder with radius r . As in Maddocks and Keller, assume that the ground terrain is modeled as a locally

smooth surface, and can be represented by surface M' [73]. Moreover, assume that the rope is represented as a curve along its centerline, and that it has an arc-length parametrization s . The curve $\mathbf{x}(s)$ describes the center of the rope, which lies on a surface M that is parallel to the ground surface M' , at a distance r above it.

The tension of the rope is given by the function $\mathbf{T}(s)$. This function is always positive and only acts in the direction $\hat{\mathbf{t}}$:

$$\mathbf{T}(s) = T\hat{\mathbf{t}} > 0. \quad (3.6)$$

One could interpret this as a constitutive assumption: the internal force exerted by tension will always be tangent to the centerline, because the rope is unable to support shear stress.

Note that forces and masses will be written in terms of densities, or the total force or mass per unit of rope.

3.2.3 Force Balance Equations from First Principles

Maddocks and Keller take what is perhaps the more intuitive approach to writing down the force balance equations for a rope on a surface [73]. Let $\mathbf{I}(s)$ be the net internal force, obtained by integrating stresses across each orthogonal cross section. From Subsection 3.2.2, that internal force is given by $\mathbf{I}(s) = T(s)\hat{\mathbf{t}}$, meaning that

$$\mathbf{I}(s) = \int_A \sigma dA = T(s)\hat{\mathbf{t}}. \quad (3.7)$$

The internal force density, or force per unit length, is $\frac{d\mathbf{I}(s)}{ds}$. If $\mathbf{E}(s)$ represents all the external forces acting on the rope per unit length, the force density balance equation is

$$\frac{d\mathbf{I}(s)}{ds} = \mathbf{E}(s). \quad (3.8)$$

In other words, the change in tension per unit length is equal to all the external forces acting on that rope per unit length.

The other forces acting on the rope include gravity, friction, and the normal force due to the surface. Let $\mathbf{W}(s) = (W_t, W_n, W_g)$ denote the external forces per unit length, or the external force densities, such as that due to gravity. Let $\mathbf{F}(s) = (F_t, F_n, F_g)$ denote the friction forces per unit length in the $\hat{\mathbf{t}}$, $\hat{\mathbf{n}}$, and $\hat{\mathbf{g}}$ directions, respectively. Therefore, $\mathbf{E}(s) = F_t\hat{\mathbf{t}} + F_n\hat{\mathbf{n}} + F_g\hat{\mathbf{g}} + \mathbf{W}$ and Equation (3.8) becomes

$$\frac{d(T(s)\hat{\mathbf{t}})}{ds} = F_t\hat{\mathbf{t}} + F_n\hat{\mathbf{n}} + F_g\hat{\mathbf{g}} + \mathbf{W}. \quad (3.9)$$

Expanding $\frac{d(T(s)\hat{\mathbf{t}})}{ds}$ and applying Equation (3.4) for normal and geodesic curvature, the left side of Equation (3.9) yields

$$\frac{dT}{ds}\hat{\mathbf{t}} + T\frac{d\hat{\mathbf{t}}}{ds} \cdot \hat{\mathbf{n}} + T\frac{d\hat{\mathbf{t}}}{ds} \cdot \hat{\mathbf{g}} = \frac{dT}{ds}\hat{\mathbf{t}} + T(\kappa_n\hat{\mathbf{n}} + \kappa_g\hat{\mathbf{g}}). \quad (3.10)$$

As described in Subsection 3.2.1, $\kappa_g(s)$ is the geodesic curvature and $\kappa_n(s)$ is the normal curvature. Substituting Equation (3.10) back into Equation (3.9) results in the following force balance equation:

$$\frac{dT}{ds}\hat{\mathbf{t}} + T(\kappa_n\hat{\mathbf{n}} + \kappa_g\hat{\mathbf{g}}) = F_t\hat{\mathbf{t}} + F_n\hat{\mathbf{n}} + F_g\hat{\mathbf{g}} + \mathbf{W}. \quad (3.11)$$

This equation can be divided into components along the $\hat{\mathbf{t}}$, $\hat{\mathbf{n}}$, and $\hat{\mathbf{g}}$ directions:

$$\begin{cases} \frac{dT}{ds} = F_t + W_t, \\ \kappa_n T = F_n + W_n, \\ \kappa_g T = F_g + W_g. \end{cases} \quad (3.12)$$

Let $F_c(s)$ denote the frictional force on the rope due to the ground, while the rope is in static equilibrium. Using the Coulomb model of friction, $F_c \leq \mu F_n$, the frictional forces on the rope are

$$F_t^2 + F_g^2 \leq \mu^2 F_n^2. \quad (3.13)$$

Recall that μ here is the coefficient of static friction; the Coulomb friction model in Equation (3.13) describes the force limits needed to maintain static equilibrium.

Substituting Equation (3.12) into the Coulomb friction model in Equation (3.13) results in

$$\left(\frac{dT}{ds} - W_t\right)^2 \leq \mu^2(\kappa_n T - W_n)^2 - (\kappa_g T - W_g)^2. \quad (3.14)$$

Condition 3.14 must be satisfied if the rope rests in static equilibrium [73]. If this condition is satisfied and $F_n > 0$, the rope sticks to the ground [74]. A version of this equation will be used to check if a given patch of ground acts as a tether friction obstacle, given the tether configuration and tension.

3.2.4 Equilibrium Conditions with a Weightless Tether

Konyukhov [74] assumes a weightless tether, $\mathbf{W}(s) = 0$, so that the force balance equation in Equation (3.14) becomes

$$\left(\frac{dT}{ds}\right)^2 \leq \mu^2(\kappa_n T)^2 - (\kappa_g T)^2. \quad (3.15)$$

Rearranging yields the full Coulomb's friction model equation for a weightless tether, which will be used in Section 3.3:

$$\left(\frac{dT}{ds}\right)^2 \leq (\mu^2 \kappa_n^2 - \kappa_g^2) T^2. \quad (3.16)$$

The term inside the parentheses must be positive; otherwise, taking the square root of the equation to find $\frac{dT}{ds}$ would result in an imaginary term on the right hand side. Since $\mu^2 \kappa_n^2 - \kappa_g^2 > 0$, the following must be true in order for a tether to stick to a surface in static equilibrium:

$$\omega^2 \triangleq \mu^2 \kappa_n^2 - \kappa_g^2 > 0. \quad (3.17)$$

This is significant because $\omega(s)$ encapsulates almost everything about the tether-terrain interaction: the friction coefficient, the shape of the terrain, the shape of the tether, and how it's lying on the terrain. Only tension and the change in tension over the tether are excluded. Therefore, Equation (3.17) can be useful in determining if static equilibrium is even possible, at any tension.

Using Equation (3.5), the previous equation can be restated as

$$\mu^2 \kappa^2 \cos^2(\phi) - \kappa^2 \sin^2(\phi) > 0, \quad (3.18)$$

which simplifies to

$$\mu^2 > \tan^2 \phi, \quad (3.19)$$

where ϕ is the angle between $\hat{\mathbf{N}}$ and $\hat{\mathbf{n}}$, as illustrated by Figure 3.3.

Knowing the friction coefficient, one can easily check if any given tether shape has the *potential* to stick, based on the angle between its Frenet-Serret normal and the surface normal. This is further detailed in Subsection 3.3.2. Note that this in no way ensures that sticking will occur. Even if Equation (3.19) is satisfied, sticking still depends on tether tension, as in Equation (3.14) [74].

3.2.5 Special Cases: Capstan Equation and Geodesics

Both Konyukhov and Maddocks and Keller show that the weightless-tether equation resulting from Equation (3.16) are generalizations of the the capstan equation [73] [74]. Although the capstan equation is not used in my motion planning algorithm, it is useful as a proof of concept and as a starting point in conceptualizing ropes on surfaces. Mechanical engineers may be familiar with this equation from textbooks and mechanics classes.

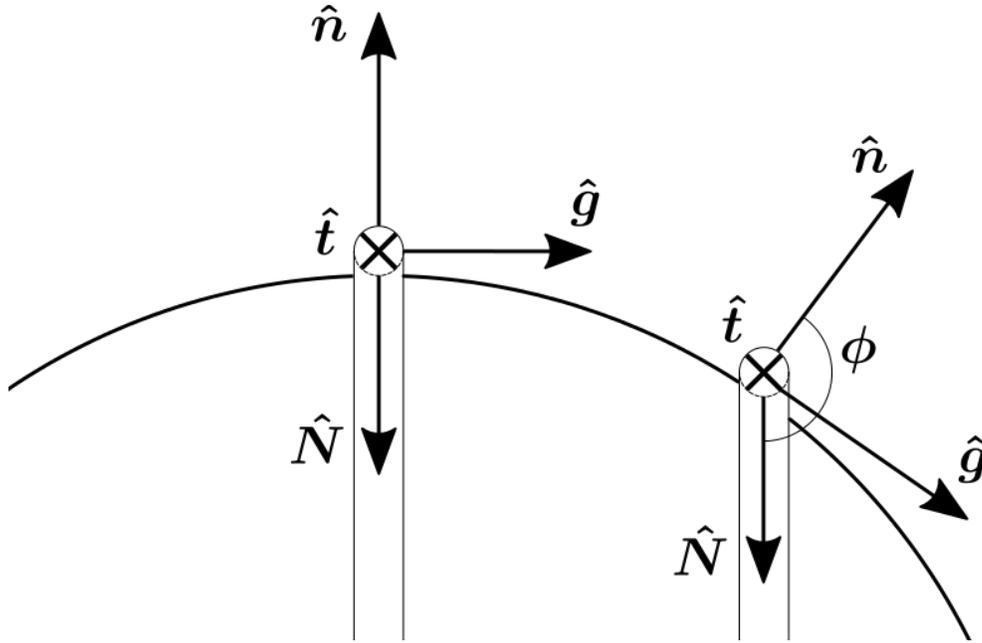


Figure 3.3: Ropes on a frictional sphere, illustrating the effect of ϕ on static equilibrium. When the rope is lying across the sphere's apex, $\phi = \pi$ and the rope is unlikely to slip. At a lower ϕ , however, the rope may slip.

Given a weightless rope wrapped around a capstan, bollard, or other frictional cylinder, the capstan equation, also known as the Euler-Eytelwein formula, relates the tension on one end of the rope (T_{load}) to the tension required to hold the system in static equilibrium (T_{hold}). ζ is the effective contact angle, or the angle over which the rope is touching the capstan.

$$T_{load} = T_{hold}e^{\mu\zeta}. \quad (3.20)$$

According to the capstan equation, the force of a load that can be held without slipping increases exponentially as the contact angle of the rope increases. Note, though, that the (positive) sign of the $\mu\zeta$ term is just convention. One could just as easily solve for the tension required to hold the load, $T_{hold} = T_{load}e^{-\mu\zeta}$. The sign of the exponential term and which way the tension changes depend on how the tension terms are labeled.

Konyukhov shows that the capstan equation is a special case of Equation (3.16) [74]. Equation (3.16) can be rearranged as

$$-\omega T < \frac{dT}{ds} < \omega T \quad (3.21)$$

and integrated so that

$$\int_s -\omega ds < \int_T \frac{1}{T} dT < \int_s \omega ds. \quad (3.22)$$

If ω is constant, which is true for the constant curvature of the rope on a capstan,

$$-\omega s < \ln T|_{T_1}^{T_0} < \omega s, \quad (3.23)$$

which becomes

$$e^{-\omega s} < T_1/T_0 < e^{\omega s}. \quad (3.24)$$

In the case of a rope wrapped around a capstan with radius R , the rope's curvature is the same as the capstan's, $\kappa = 1/R$. All of that curvature is in the direction normal to the capstan's surface, so $\kappa_n = 1/R$ and $\kappa_g = 0$. By Equation (3.17), $\omega = \pm\mu\kappa_n = \pm\frac{\mu}{R}$.

To put s in terms of ζ , consider that the circumference of a circle is $2\pi R$, and the length of an arc of a circle is ζR , where ζ is the angle that arc subtends. Therefore, Konyukhov's ωs term is equivalent to $\pm\frac{\mu}{R}\zeta R = \pm\mu\zeta$. Assuming that $T_{load} > T_{hold}$, like the capstan equation does, selects for the positive $\mu\zeta$. Thus, Equation (3.20) is a specialized case of Equation (3.16).

It is important to note that the capstan equation, like Equation (3.16), provides only a tension *limit* for which the system is in static equilibrium. The loading tension T_{load} can always be lower; this will result in a smaller effective contact angle. These equations cannot be used to calculate an exact tension on a tether, only to find a limiting tension past which the tether will slip.⁶

In fact, this variation on the capstan equation in Equation (3.24) describes any situation in which ω is constant with respect to s . This occurs when κ_n and κ_g are constant; in other words, the curvature in both the normal and geodesic directions is constant.

As was stated in Subsection 3.2.1, a curve is a geodesic (on a surface S) if $\kappa_g = 0$. In fact, according to Maddocks and Keller, "the geodesics are the curves of maximal growth or decay of tension. Moreover, in the absence of friction, they are the only equilibria with T not equal 0" [73]. Konyukhov shows the converse, that "the equilibrium curve on a rough surface under maximum tension load is geodesic." [74] This means that, for a weightless tether, as tension approaches infinity (or

⁶Konyukhov provides a system of equations to describe the sliding contact of a tether on a surface, but they are highly non-linear (except for in certain special cases) and are outside the scope of this work [74].

friction approaches 0), the tether will deform into a geodesic. Since a geodesic is a locally shortest path, at high tension the tether will tend toward the SHP.

3.3 Computing Tether Tension and Static Equilibrium

The examples in this section use the equations from Section 3.2, and form the basis for the algorithms in Section 3.4.

3.3.1 Model and Assumptions

The conditions for static equilibrium that stem from Equation (3.16) depend on the assumption of a weightless tether. Fortunately, this is a reasonable assumption in the case of Axel's tether, which was specifically designed to be tough but lightweight. The current tether has a mass of 15 g /m, which over 100 m adds up to a mere 1.5 kg — a fraction of Axel's approximately 70 kg mass. At 1 km, a goal distance for Axel, the 15 kg of tether might be significant, if the tether were actually in the air the whole time. At such a distance, however, Axel is likely to have passed multiple anchors or changes in slope, on which the tether can rest. Besides acting as capstans (and thus reducing the load on the initial anchor point, should Axel slip and suddenly place high tension on the tether), these anchors ensure that at any given time, only a small length of tether will be free to move.

Furthermore, Equation (3.16) will be used to identify tether friction obstacles. These TFOs will generally be significant when Axel is climbing, with the tether held taut at high tension. Even when not moving, the tether management system⁷ should ensure that the tether never goes slack. This is significant because tether weight has less impact as tension increases. Consider the terms in Equation (3.14). The normal force component (that gets multiplied by μ) is $(\kappa_n T - W_n)^2$. At sufficiently high tension, the $\kappa_n T$ term dominates the W_n term. The same is true for the geodesic or dragging force component, $(\kappa_g T - W_g)^2$. The force component in the tangent direction is $(dT/ds - W_t)^2$, so a high change in tension $\frac{dT}{ds}$ would make tether weight negligible.

Experimental tension measurements also show that, at typical driving tensions, the tether weight contributes only a small fraction of the forces on the tether. During a 2011 field test near Black Point Lava Flow in Arizona, the Axel team found that tether tension reached a maximum of 170 lbs-force (756 N) for a then-55-kg robot descending 20-80° slopes [58]. 100 m of tether would exert almost 12 N of force in

⁷Axel's tether management system is described in Section 1.5.

the direction of gravity, or 1.5% of the tether tension force.

Modeling tether friction against the terrain naturally depends heavily on the coefficient of (static) friction. In his paper on tether friction in rope rescue, Attaway's tables use a range from $\mu = 0.25$ to 0.45 , and he chooses $\mu = 0.4$ for one example [75]. Unless noted otherwise, I have likewise used $\mu = 0.4$. Because μ changes so much depending on the surface, however, the frictional tethered planner can only provide a first-order approximation of the true situation. It should be used as an aid to an operator, and not be relied upon as the sole determinant that a path is safe to travel.

Unfortunately, the wide range of μ for different surfaces also means that estimating the friction coefficient will probably be the hardest step in frictional tethered planning in the field. Camera images alone will not suffice. The easiest method might be to test the coefficient of friction for a wide range of materials in the field, by scraping the surface with an arm or wheel equipped with a force sensor. Enough of these tests would allow an operator to match a camera image of a dirt type with an experimentally-determined μ for a similar-looking Martian or lunar dirt.

3.3.2 Conditions for Sticking at a Friction Anchor

The equations in Section 3.2 allow one to determine whether a tether will stick (i.e., remain in static equilibrium) on a surface. However, Equation (3.16), the weightless Coulomb's law equation in Subsection 3.2.4, is an ordinary differential equation (ODE) for tether tension. This can be expensive to compute, especially for the combinations of tether configurations and tensions at many possible points. Instead, it behooves us to use Konyukhov's conditions for static equilibrium to first narrow the field of possibilities.

According to Konyukhov, the following 3 conditions must be satisfied for a weightless rope to lie in static equilibrium on a surface [74]:

1. There is contact between the rope and the surface, with no separation. F_n is positive for all points of the curve:

$$F_n = -\kappa_n T > 0. \quad (3.25)$$

2. The friction coefficient and angle ϕ satisfy the following for all points of the curve:

$$-\mu < \tan \phi < \mu. \quad (3.26)$$

3. The tension forces at both ends, T and T_0 , satisfy the following:

$$T_0 e^{-\int_s \omega ds} < T < T_0 e^{\int_s \omega ds}. \quad (3.27)$$

Since $T > 0$, Equation (3.25) is equivalent to $\kappa \cos \phi < 0$. In addition, $\kappa(s) > 0$, which requires that $\cos \phi < 0$. Thus, for Equation (3.25) to be true, $\pi/2 < \phi < 3\pi/2$. In Figure 3.3, this means that $\hat{\mathbf{N}}$ must fall in the half-space formed by the $(\hat{\mathbf{t}}, \hat{\mathbf{g}})$ plane, on the side opposite $\hat{\mathbf{n}}$. Condition 1 in Subsection 3.3.3 will show what should be intuitively obvious: that a tether cannot stick along a surface that is concave in the direction of its tangent vector.

Equation (3.26) is also a restriction on ϕ , which is analogous to a friction cone constraint in robotic manipulation and grasping.⁸ As is depicted in Figure 3.3, the friction coefficient μ determines the width of a cone extending opposite from the surface normal, $\hat{\mathbf{n}}$. If the Frenet-Serret normal $\hat{\mathbf{N}}$ in the direction of the curve's acceleration falls inside this cone, sticking is possible for certain tether tensions. If it does not, the rope will certainly slip.

The surface convexity requirement resulting from Condition 1 can be tested for any surface, although for certain points whether a tether can stick will depend on the direction of the tether's tangent. Terrain data can be preprocessed in this way before the motion planning algorithm is run, to speed up the process. Fully testing Conditions 1 and 2, though, requires a tether configuration.

The planning algorithm to be described in Section 3.4 will provide a potential SHP, representing a taut tether in a known configuration. From that, the tether's curvature, κ , and angle between normals, ϕ , can easily be calculated. If Condition 1 holds true, Condition 2 can be checked with the same variable ϕ , as well as the friction coefficient μ . If Condition 2 is true, sticking is theoretically *possible* for this tether configuration.

Condition 3 depends on tether tension and the change in tether tension. While one could use Equation (3.16) to check if a tether will slip, this requires the calculation of $\frac{dT}{ds}$, a difficult term to estimate, or measure in the field. However, by making some assumptions about the average and limits of Axel's driving tether tension, one can estimate the tension on either end of a tether around a PTFO, T and T_0 . Subsection 3.4.1 covers these assumptions in more detail.

⁸This is a different friction cone from that given for Axel's traversability in Chapter 2.

3.3.3 Analysis of Surface Features

For a weightless tether to maintain its shape on a frictional surface, the surface must be convex in that direction. Principal directions of surface curvature, which are easily computed at each point, can be used to rule out certain spots on the surface as being likely to cause tether slippage. The reasoning for this can be seen from the definitions below.

Consider a given surface M at point p . The principal curvatures κ_1 and κ_2 are the maximum and minimum values for curvature at that point; they quantify how the surface bends in different directions. The directions in the normal plane in which curvature is at its maximum and minimum are called the principal directions, $\hat{\mathbf{v}}_1$ and $\hat{\mathbf{v}}_2$, and they form an orthonormal basis as long as $\kappa_1 \neq \kappa_2$. At any point, a vector \mathbf{u} can be defined by an angle ξ with respect to the principal directions such that $\mathbf{u} = \cos(\xi)\hat{\mathbf{v}}_1 + \sin(\xi)\hat{\mathbf{v}}_2$.⁹

Although Section 3.2 generally discussed κ_n in regard to a curve, it can also be described for a vector tangent to that curve.

Theorem 3.3.1 (Meusnier's Theorem) *All curves lying on a surface M passing through a point p on M with the same tangent line have the same normal curvature [98].*

As a result, κ_n can be defined in terms of the principal curvatures. If a curve's tangent vector is represented by a vector \mathbf{u} at an angle ξ with respect to $\hat{\mathbf{v}}_1$,

$$\kappa_n = \kappa_1 \cos^2(\xi) + \kappa_2 \sin^2(\xi). \quad (3.28)$$

Alternately, the principal directions can be thought of as the angles of ξ for which κ_n is at a maximum and minimum:

$$\kappa_1 = \max_{\xi} \kappa_n(\xi), \quad (3.29)$$

and

$$\kappa_2 = \min_{\xi} \kappa_n(\xi). \quad (3.30)$$

⁹Principal curvatures and principal directions are the eigenvalues and eigenvectors, respectively, of the shape operator at that point. The shape operator is a symmetric linear map $S_p : M_p \rightarrow M_p$. $S(\mathbf{v}) = -\nabla_{\mathbf{v}}\hat{\mathbf{n}}(p)$; that is, the shape operator is the negative derivative of the unit normal of the surface.

Recall that by Equation (3.25), static equilibrium requires $\kappa_n > 0$. Any wholly concave surface will have $\kappa_1, \kappa_2 < 0$, meaning that κ_n can never be positive. If $\kappa_1, \kappa_2 > 0$, Equation (3.25) is satisfied, and the program can move on to the next step in checking for possible sticking. If one of the principal curvatures is greater than 0 and the other is less than 0, p is a saddle point. In this case, tether sticking is only possible at angles ξ for which $\kappa_n > 0$.

Using Equation (3.28), the following equation determines the break point ξ_{flat} at which $\kappa_n = 0$:

$$\xi_{flat} = \tan^{-2} \left(\frac{-\kappa_1}{\kappa_2} \right). \quad (3.31)$$

This equation can be used to determine the range of angles for which sticking is possible.

3.3.4 Numerical Solutions for Tension

Although there is no general closed form analytical solution to Equation (3.14), it can be solved numerically for a given initial tension T_0 . The examples in this section were computed in Matlab, using Matlab's ode45 solver. Unless otherwise stated, they were calculated with a friction coefficient of $\mu = 0.4$.

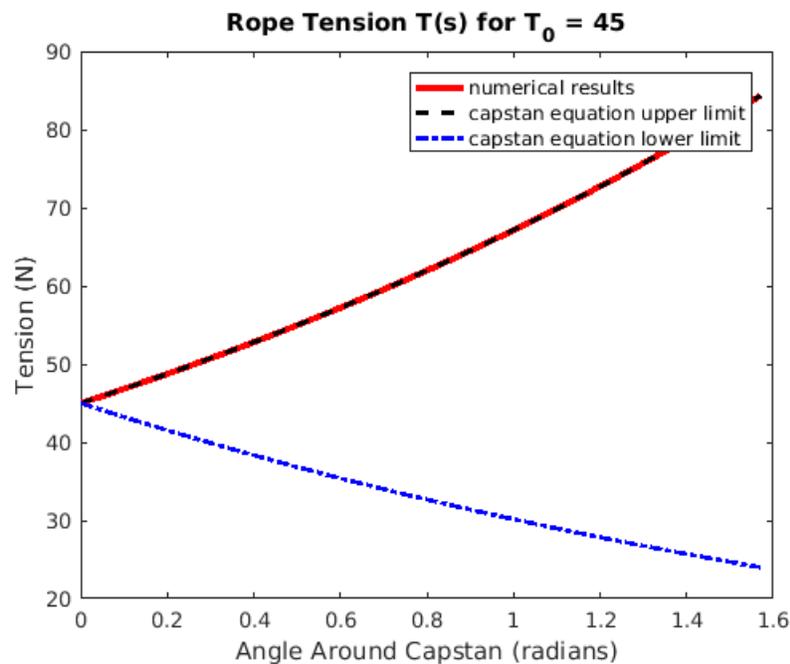


Figure 3.4: Numerically computed tension along a capstan matches the well-known analytic solution, increasing or decreasing exponentially over the contact angle from 0 to $\frac{\pi}{2}$ radians.

Figure 3.4 shows the numerically-computed tension along a rope wrapped around a capstan, from 0 to $\frac{\pi}{2}$ radians. Recall that the capstan equation gives both an upper and lower limit to the tension at the end of the rope, depending on which end tension is considered T_{load} and which is considered T_{hold} .¹⁰ Those tension limits are shown by the dashed black and blue lines. The solid red line represents the numerically-computed tension with the same assumptions as the capstan equation: the rope is weightless, $\kappa_g = 0$, and $\kappa_n = 1/R$ where R is the radius of the capstan. Although $R = 0.5$ m here, examination of the capstan equation will show that the capstan's radius does not matter, only the angle contacted by the rope. As expected, the numerical solver computes the same tension limit as the capstan equation (whether that is an upper limit or a lower limit depends on how one sets up the inputs, and can easily be changed).

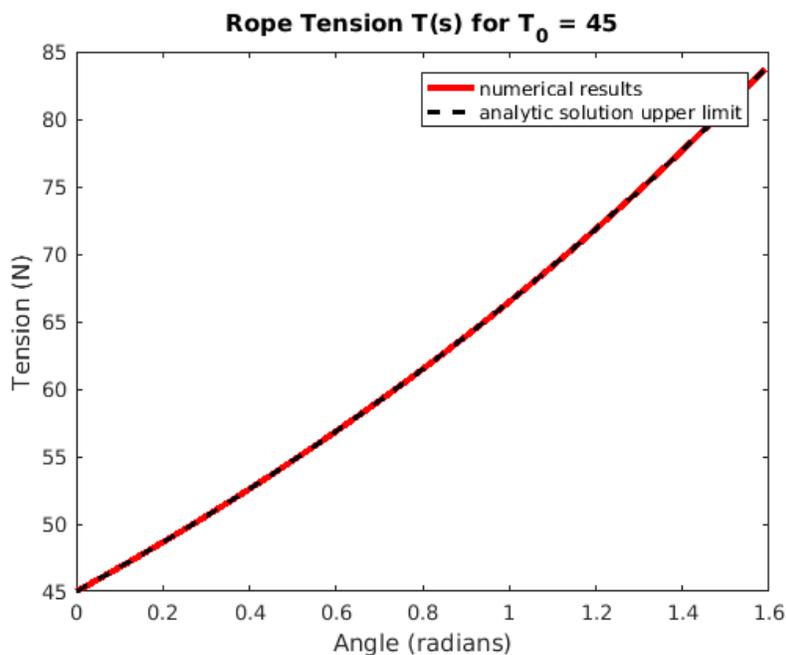


Figure 3.5: Tension along a helix with a radius $R = 1$, pitch $H = 1$, and starting tension $T_0 = 45$ N.

The slightly more complicated setup used in Figure 3.5 also serves as a test case, comparing Konyukhov's analytical solution to the numerically computed tension in a weightless rope wrapped in a helix around a cylinder. That helix is a geodesic, so $\kappa_g = 0$. In this example, the cylinder has a radius of $R = 1$ m, and a pitch of $H = 1$ m, describing the height in z for the rope to make 1 full rotation. The figure

¹⁰For more explanation, see Subsection 3.2.5.

shows tension for 0 to $\pi/2$ radians. Once again, the numerical solution matches the analytic one.

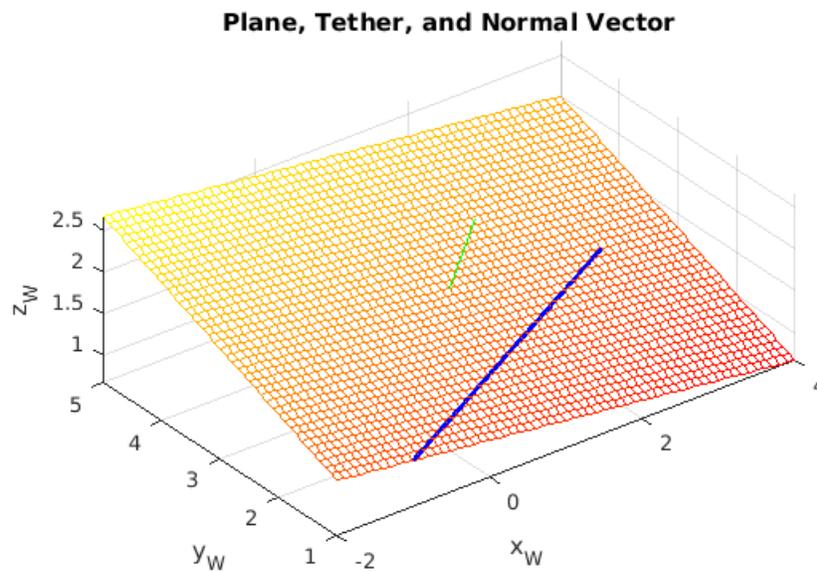


Figure 3.6: A heavy rope segment (blue) lying on a titled plane, whose surface normal vector is shown in green.

Figure 3.6 depicts the geometry of the next example, for a heavy rope on a tilted plane. Unlike in the previous examples, there is no analytical solution, since the weightless rope approximations are no longer valid. In this example, the rope has a mass density of 100 g/m , which is almost 7 times the mass density of Axel's current 15 g/m tether. The numerical results for tether tension over the tether are shown in Figure 3.7. Unlike in the previous examples, both κ_n and κ_g are 0, so only the rope's weight contributes to a change in tension. In fact, this example is a variation on the common physics-textbook problem: if a chain is suspended in the air from one end, what is the tension along the chain? In both cases, the tension changes solely due to the weight of the chain or rope, which increases along the length of that rope.

The next set of experiments examined both tether sticking and tether tension on a slope. The problem's geometry is shown in Figure 3.8. A heavy tether (black) is anchored at the top of a shallow slope, and continues over an edge to a much steeper slope. The bottom of the tether is loaded with a tension of 539 N , the force provided by Axel's 55-kg -mass due to Earth gravity, hanging straight down (conceptually, imagine the lower slope ends in an overhang, and Axel is hanging

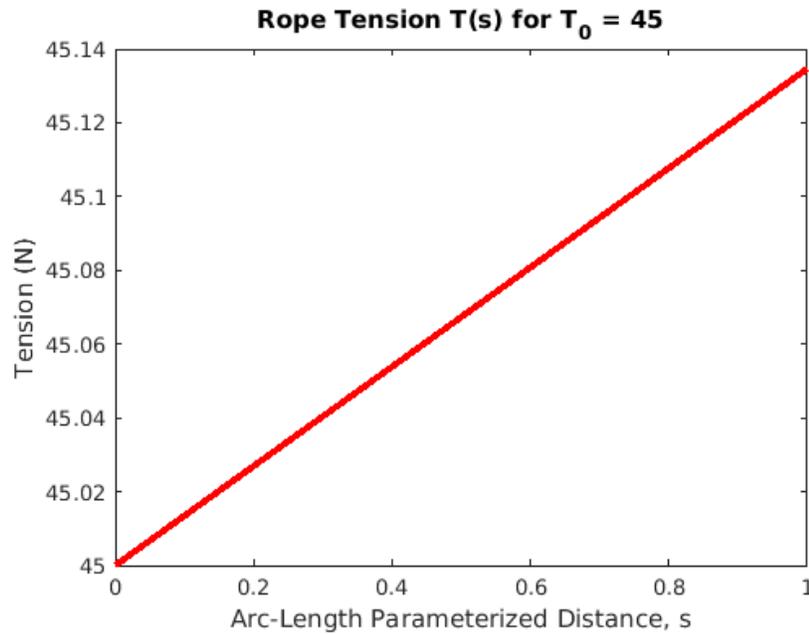


Figure 3.7: Numerically computed rope tension for a heavy (0.1 kg/m) rope on a tilted plane.

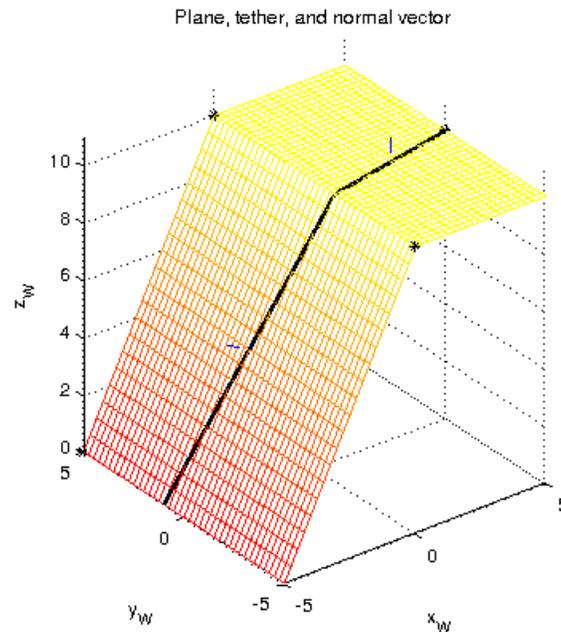


Figure 3.8: A heavy tether is anchored to a shallow slope, followed by a steeper slope, in the geometry shown here.

down unsupported, below). Tether length is measured from the hanging-robot end, where $s = 0$. The questions to be investigated were twofold: would the tether stick

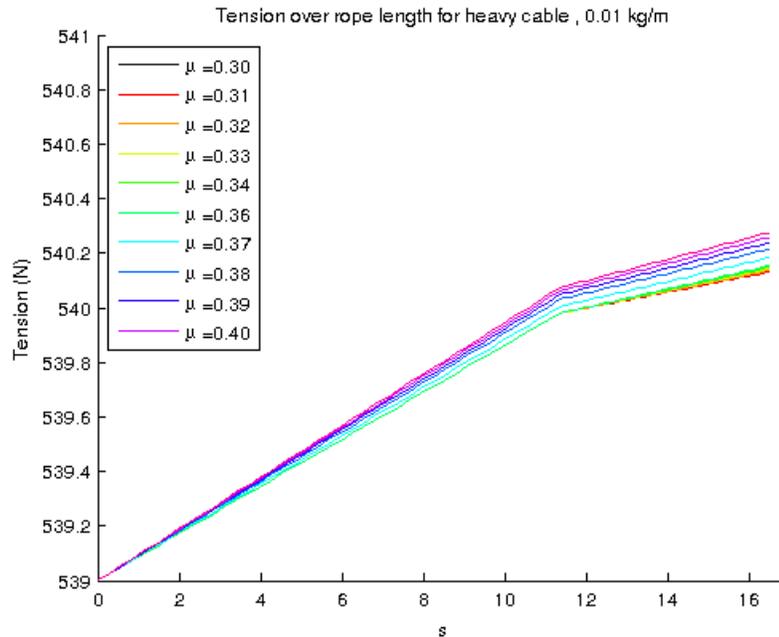


Figure 3.9: The heavy tether shown will stick to the shallower slope with a low friction coefficient, but will only stick to the steeper slope at higher friction coefficients. μ also affects how quickly tension changes along the tether.

(stay in static equilibrium) in these conditions? And if it did, how would the tension on the tether change?

The graph in Figure 3.9 shows how tension changes over the length of the tether, and how sticking depends on the coefficient of friction. Each colored line represents the rope tension for that coefficient of friction; if the line is not shown, the tether did not stick to the surface under those conditions. For example, the tether sticks to the shallower slope at $\mu = 0.31$, and results in a lower tension change than that for the higher friction coefficients. On the steeper slope, however, the tether with $\mu = 0.31$ will slip sideways. Only the higher friction coefficients are sufficient for sticking at this slope.

One result of the capstan effect is that it allows even loose initial anchors to hold over a long traverse. The anchor must hold out against the initial tension as Axel drives away, of course. However, after the tether wraps around a few anchors (either sideways, or over hilly friction anchors), that initial anchor becomes insulated against any slips or falls that might result in higher-than-expected tensions.

3.3.5 Checking if Sticking is Possible

Before testing whether a particular tether configuration will stick, terrain data can be preprocessed to eliminate all the concave areas where the tether will not stick. As described in Subsection 3.3.3, Condition 1 (Equation (3.25)) can be checked by finding the principal curvatures at each point in the terrain. The example below uses a small patch of terrain data from Shackleton crater on the Moon, as shown in Figure 3.10.

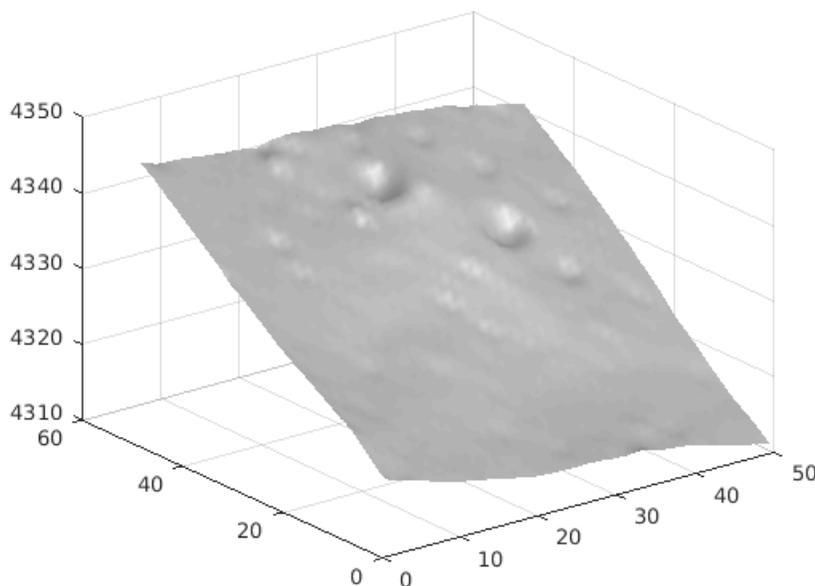


Figure 3.10: A small surface patch of Shackleton crater on the Moon.

In Figure 3.11, convex points are given in green and concave points in red. The concave points can be immediately eliminated; a tether through one will never stick there. At the saddle points in blue, however, the surface is convex in one direction but concave in another. Since whether a tether can stick depends on the angle of the tangent vector through those points, Figure 3.12 additionally shows those vectors through saddle points for which sticking is possible. Once again, convex points are in green and concave in red. Saddle points are shown in black, overlaid by cyan arrows in every direction for which the surface is convex.

Figure 3.13 checks Conditions 1 and 2, to determine if sticking is possible at each point along the tether. Points that will slip are shown in red, and points that *may* stick, with the right tension, are shown in black.

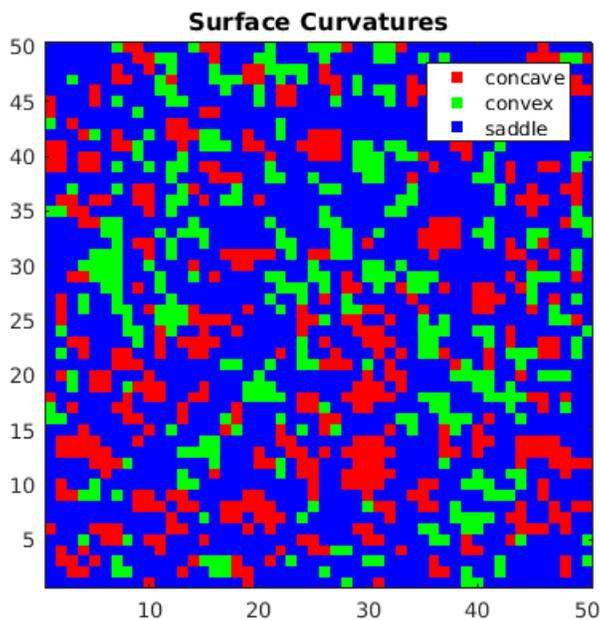


Figure 3.11: Convexity of points on the surface.

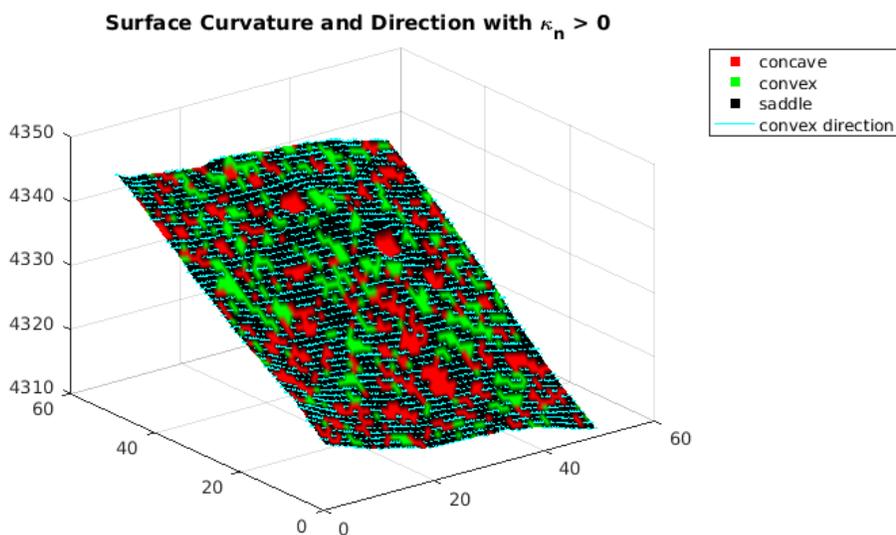


Figure 3.12: Convexity of points on the surface, with arrows showing convex directions for saddle points.

In Figure 3.14, the previous two tether-configuration conditions are checked, as well as the tension on the tether. In this case, sticking is possible; Figure 3.15 shows that ϕ meets the $\arctan \mu$ constraints, for $\mu = 0.4$, near the middle of the tether. However, a sufficiently high tension differential will cause the tether to slip, violating Condition

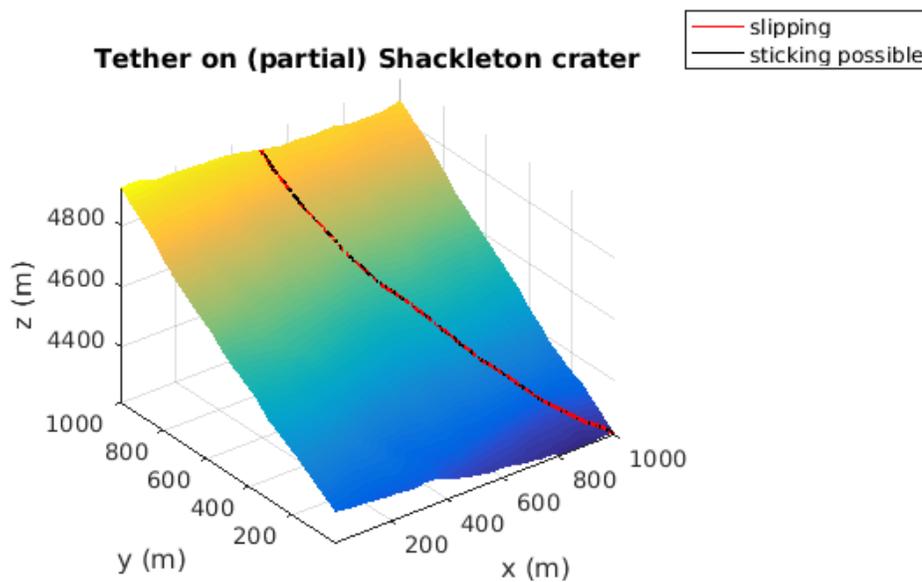


Figure 3.13: A tether on a section of Shackleton crater. Places where sticking is possible are marked in black.

3.

3.4 Frictional Tether Planning

As the preceding sections have shown, determining whether a frictional tether will slip on terrain requires the terrain shape and friction coefficient, the tether configuration, and the tension at either end of the tether. Algorithm 7 generates a test SHP, which it uses as a tether configuration to test if a PTFO will hold.

3.4.1 Assumptions about Tether Tension

One common misconception about Axel's tether configuration is that it will take the same shape as Axel's path. While this could happen for a robot trailing its slack tether on a tether-free plane, this is not generally the case for Axel. Instead, because Axel's tether management system will attempt to keep the tether taut, it is reasonable to assume that $T > 0$.¹¹ That taut tether will always take the shortest path back to the proximal anchor point, regardless of how circuitous Axel's path may be. This

¹¹This is true in any static approximation. Dynamically speaking, it is possible for the tether to suddenly go slack — perhaps by losing the anchor — and the motors will take some non-zero time to draw the tether tight again.

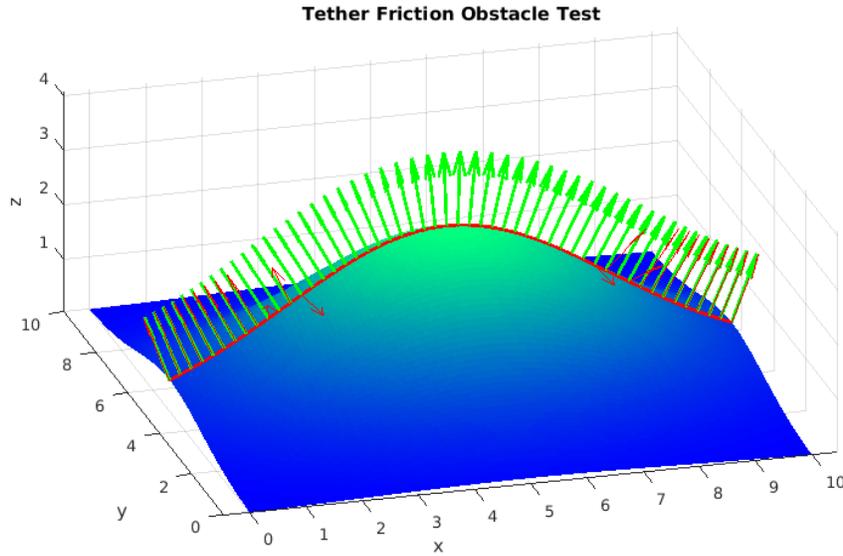


Figure 3.14: An example of a tether on a friction anchor, or a convex patch of terrain. Although the tether (in red) appears to go directly over the hill's apex, it is actually slightly off-center. The red vectors represent the tether normals, or Frenet-Serret normals, while the green vectors are the surface normals.

is true even if the taut tether catches on the ground, in which case that terrain area is a TFO and its tether contact point is the new proximal anchor. Because of this behavior, the tether position when Axel is at v_g after a descent will be the SHP, and after ascent the tether will be entirely reeled up. At any point partway down, the tether configuration will look like the SHP, up to the proximal anchor point.

Let T_A denote the Axel-side tether tension, measured where the tether enters Axel's arm. Given knowledge of Axel's geometry and the slope, T_A can be estimated. Abad-Manterola gives the following equation for tether tension [58]:

$$T_A = \frac{m g r_w \sin \eta_i}{L_c \sin \theta_c + \eta_i + \theta_t + r_w \cos \eta_i - \theta_t}, \quad (3.32)$$

where m is Axel's mass, r_w is the wheel radius, L_c is the arm length, η_i is the slope angle, θ_t is the tether angle, and θ_c is the angle of the arm [58].

Let T_0 denote the tension just after (downhill of) the initial anchor point a_0 , and T_i the tension after anchor a_i . Assume that the tether tension does not change appreciably between anchor points. The tether may lie lightly on the ground between anchors, or span the distance between them in the air. In either case, in the absence of κ_n

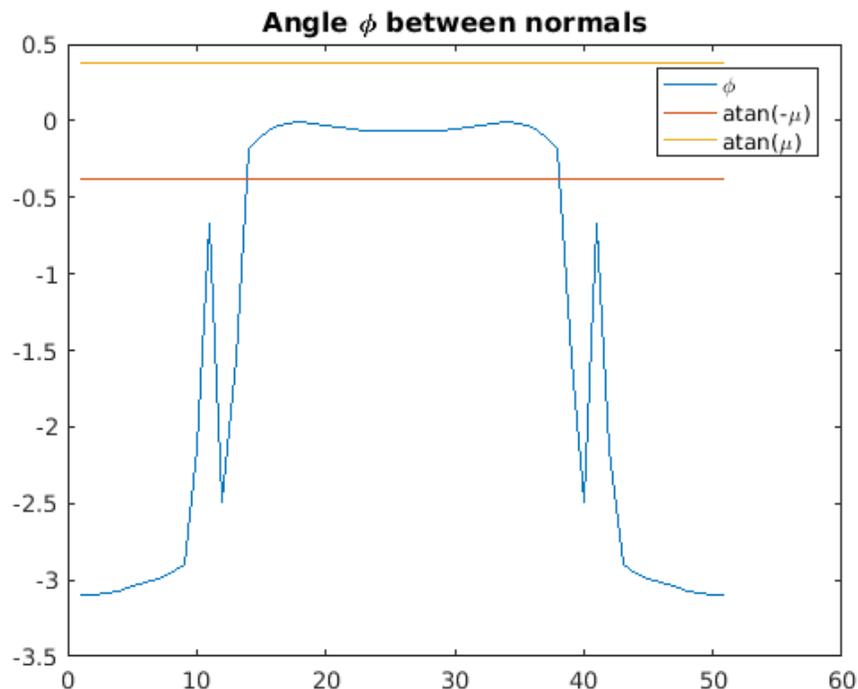


Figure 3.15: The angle ϕ from Figure 3.14, plotted over tether length. While the tether is at the top of the hill, ϕ is between the friction constraints, meaning that sticking is possible there.

and κ_g components the only contributor to the term $\frac{dT}{ds}$ is the tether weight, as in Figure 3.7. Unlike in that example, though, Axel's tether is very light, so its weight can be ignored. Therefore, it is a reasonable assumption that the tension just above anchor a_i will be equal to T_{i-1} .

To gain some intuition about how tether tension changes over time, depending on $T_A(t)$, consider the following hypothetical situation. Imagine Axel driving on a tether-demand plane, which implies a positive but likely time-varying tether tension, $T_A(t)$. As it leaves a_0 at time $t_0 = 0$, $T_0(t_0) = T_A(t_0)$. Axel then moves down past anchor a_1 , where the tether makes contact at time t_1 . Right before that happens, $T_0(t_1 - \epsilon) = T_A(t_1 - \epsilon)$. Unless the tether slips, T_0 will now stay constant at $T_A(t_1 - \epsilon)$. After making contact with the anchor, $T_1(t_1) = T_A(t_1)$. Now, like in the situation described by the capstan equation, Axel can increase or decrease its tension within certain limits without the tether slipping at a_1 . Because of capstan effect, different segments of tether may stay at different tensions, especially if $T_A(t)$ varies wildly during Axel's movement.

While it is possible to simulate tension everywhere along the tether, using this logic and possibly measurements of $T_0(t)$, the process would be prohibitively computation-intensive to include in a friction obstacle search. Instead, use some conservative estimates to group PTFOs into categories of solid TFOs, never-obstacles, and those obstacles whose sticking/slipping is too unreliable to be depended on. First, assume the maximum and minimum expected tensions are known, T_{max} and T_{min} . For example, these could readily be computed from Axel's mass, the gravitational constant, and the steepest and shallowest slope angles that Axel is likely to experience on a given exploration campaign. Alternately, one could estimate these constants from experimental tether tension measurements, such as those taken during Axel's field testing in Black Point, Arizona [58]. Assume, too, that because of its tether management system Axel usually drives with some tether tension T_d , usually within some known deviation of that, ϵ .

Based on these tensions, there are 3 basic categories for PTFOs:

1. If $T_{max} < T_{min}e^{\int_s \omega ds}$, then the tether will stick even when subjected to the most extreme tension difference that Axel could expect. This PTFO is a solid TFO; it acts like an obstacle to the tether.
2. If $T_d + \epsilon \not\leq T_d e^{\int_s \omega ds}$, then the the tether will slip at even a small tension change. (Alternately, one could just as easily use the difference between $T_d - \epsilon$ and T_d , or even $T_d - \epsilon$ and $T_d + \epsilon$.) One can expect that the normal tension changes of driving will cause the tether to slip, so this PTFO does not act as a tether friction obstacle.
3. If neither of the others is true, this obstacle could be dangerously volatile. While the tether will stick here at normal driving tensions, a sudden slip or higher tension will cause the tether to slip, resulting in a sudden loss of anchor. These should be avoided if at all possible.

The following algorithms both categorize the PTFO according to these criteria in the CheckPTFOs() function.

3.4.2 The Algorithm

Like Algorithm 6 in Subsection 2.4.3, Algorithm 7 starts with a polygon with holes, P , representing a map and its polygonal obstacles. Unlike in Subsection 2.4.3, however, those holes or obstacles must first be labeled as either geometric obstacles

Algorithm 7 Find a feasible ascent/descent path pair, with tether friction

Input: start v_s , goal v_g , polygonal map P of tether obstacles, labeled as geometric or PTFO, friction coefficient μ , tether tension cutoffs $T_{min}, T_{max}, T_d, \epsilon$

Output: (π_a, π_d) ascent and descent path pair

```

1: procedure PATH PLANNING( $v_s, v_g, P, T_{min}, T_{max}, T_d, \epsilon$ )
2:    $B \leftarrow \text{findBTM}(P)$ 
3:   global  $\mu$ 
4:    $searcher \leftarrow \text{SPTSEARCH}(v_s, v_g, B)$  ▷ Initialize search
5:   while True do ▷ Or while ( $count < cutoff$ )
6:      $\bar{\pi}_{test} \leftarrow searcher.getNextBest()$ 
7:      $(\bar{\pi}, F) \leftarrow \text{CHECKPTFOS}(\bar{\pi}_{test}, T_{min}, T_{max}, T_d, \epsilon)$ 
8:     if  $(\bar{\pi}, F)$  then
9:        $wayPts \leftarrow \text{CHECKIFFEASIBLE2}(\bar{\pi}, F)$ 
10:      if  $wayPts$  then
11:        find ascent path  $\pi_a$  through  $wayPts$ 
12:        find corresponding descent path  $\pi_d$ 
13:        return  $(\pi_a, \pi_d)$ 

```

or PTFOs. The latter can be achieved using the principal curvatures surface feature analysis described in Subsection 3.3.3. From there, triangulate P to get the boundary triangulated 2-manifold (BTM) B , as described in Subsection 2.2.2.

Like in the frictionless case, Algorithm 7 starts a search of the BTM, looking for the best SHP. In this case, though, it treats the PTFOs as regular obstacles. Having found an SHP, $\bar{\pi}_{test}$, the algorithm uses that SHP as a tether configuration to test if the tether will stick on any PTFOs, in Algorithm 8. As described in Subsection 3.4.1, this involves checking whether the tether will stick for several combinations of tensions at either end of the tether.

Algorithm 8¹² checks each friction anchor in the SHP, to determine if the tether will slip or stick. If it sticks at some tensions and slips at others, the path is considered unsafe and the algorithm returns ‘False’. If the tether slips over the obstacle, the SHP must be recomputed.

This is not a simple matter of removing that anchor f from the list. Instead, the new sleeve must be computed, and a new shortest path segment found between the anchor points on either side of f . Sometimes, that new shortest path segment will actually include more anchors, if another obstacle lies on that shortest path. Section 3.5 describes how to find that new path. Essentially, after removing the false PTFO,

¹²This algorithm was developed with the help of Dr. Jeffrey Edlund.

Algorithm 8 Check whether PTFOs act as obstacles

```

1: procedure CHECKPTFOS( $\bar{\pi}, T_{min}, T_{max}, T_d, \epsilon$ )
2:    $(A, F) \leftarrow findAnchors(\bar{\pi})$  ▷ Or store anchor type in node of  $\bar{\pi}$ 
3:    $done \leftarrow False$ 
4:   while not  $done$  do
5:      $slipping \leftarrow \emptyset$ 
6:     for  $f$  in  $F$  do
7:       if TETHERSLIPS( $f, T_d, T_d + \epsilon$ ) then
8:          $slipping \leftarrow f$ 
9:         break
10:      else
11:        if TETHERSLIPS( $f, T_{min}, T_{max}$ ) then
12:          return False
13:      if  $slipping$  then
14:         $A, F, \bar{\pi} \leftarrow UPDATEWITHSLIPPED(slipping, A, F, \bar{\pi})$ 
15:      else
16:         $done \leftarrow True$ 
17:      break
18:   return  $(\bar{\pi}, F)$ 

```

the area around it must be triangulated. One can then find the new sleeve, as a result, and can use the funnel algorithm to find the shortest path in that new sleeve.

Additionally, since a path change will alter the configuration of the tether through the surrounding anchors, those anchors must once again be checked for slipping or sticking. Once Algorithm 8 has eliminated all slipping anchors, such that the remaining anchors are geometric or are friction anchors that will definitely stick, it returns the new path $\bar{\pi}$ and new friction anchors F .

Algorithm 9 Check if tether slips at given end tensions

```

1: procedure TETHERSLIPS( $f, T_{small}, T_{big}$ )
2:   Find tether configuration at point  $f$ 
3:   Use tether configuration and  $\mu$  to find  $\phi$  and  $\omega$ 
4:   if  $-\mu < \tan \phi < \mu$  then ▷ Sticking possible
5:     if  $T_{big} < T_{small} e^{\int_s \omega ds}$  then
6:       return False
7:     else
8:       return True
9:   else ▷
10:  return True

```

Condition 1 for sticking has already been checked, in the process of finding PTFOs. Algorithm 9 checks Condition 2 and then, if it is true, checks Condition 3.

Algorithm 10 Check if SHP is Feasible

Input: a shortest homotopic path $\bar{\pi}$, and its friction anchors F **Output:** *wayPts*, if ascent feasible. False if not feasible.

```

1: procedure CHECKFEASIBILITY2( $\bar{\pi}$ ,  $F$ )
2:   for  $a$  in  $\bar{\pi}$  do
3:     if  $a$  is passable then
4:       if  $a$  in  $F$  then                                      $\triangleright$   $a$  is a friction anchor
5:         Find visibility angle  $\psi_i$ 
6:         for  $\theta$  in  $range([\pm\theta_{max}, \psi_i])$  do
7:           if not TETHERSLIPS( $a$  with angle  $\theta$ ,  $T_d$ ,  $T_d + \epsilon$ ) then
8:             return False
9:         else
10:          return False
11:   Find waypoints wayPts
12:  return wayPts

```

Even if the friction anchors are stable for the tether configuration given by the SHP, they may not stick as Axel ascends, moving so as to remove each anchor. Axel's motions will alter the tether configuration around that anchor. Therefore, Algorithm 10 must check that the tether sticks for all the tether angles in the anchor reachable set for that anchor. In the case of a friction cone, this ARS will be bounded by some angle limit θ_{max} on one side, and by the visibility line at angle ψ_i on the other, past which anchor a_i is no longer the proximal anchor.

The `CheckFeasibility2()` function must still check whether a friction anchor is passable. However, while a robot must be able to drive around the side of a geometric anchor, that same robot may drive over a friction anchor. For the point robot used in Chapter 2, this does not actually change any definitions; the anchor point must still lie inside the previous anchor's friction cone. In practice, however, a geometric anchor would require some padding around the side that a friction anchor would not. For similar reasons, the waypoints are slightly different. In Chapter 2, waypoints were chosen along an anchor's visibility line, and to the side of that anchor point. For a friction anchor, though, the anchor itself is the ideal waypoint. In fact, attempting to remove a friction anchor by driving around that tether obstacle may result in the the tether sticking on that obstacle.

3.5 Impact of Changing Obstacles

The algorithm reviewed in Chapter 2 assumed a known, stationary map of the terrain prior to the planning process. However, PTFOs introduce the possibility of

the map changing during planning, a characteristic also shared by online motion planning. In the case of the former, a PFTO may seem to disappear when Algorithm 8 determines that the tether will slip over that patch of terrain. In the case of online motion planning, the robot may receive new information as it starts down a path, forcing it to update its map to account for previously unseen or wrong obstacles. This section presents an online tethered motion planning algorithm that maintains path homotopy constraints in light of these errors and changes. It also describes how to change BTMs, sleeves, and paths when the map updates. After reviewing these concepts, the resulting addition to Algorithm 7 is given in Subsection 3.5.5.

The text in the rest of this section was modified from [53].

3.5.1 The Pre-Planning Algorithm

Our online path planning algorithm starts similarly to Algorithm 6 and Algorithm 7. We first triangulate the tether-demand plane(s) into a BTM. We then carry out an A* search, using the complex analysis concepts described in Subsection 2.2.1 to constrain that search by \mathcal{H} -value. Having found a path to the goal, we use this path to identify a sleeve, as classified by its \mathcal{H} -value. Using Hershberger and Snoeyink's algorithm, we find the SHP for the sleeve, and identify the anchor points. After checking if the desired sleeve has a feasible path, we choose ascent and descent paths from the set of feasible paths. We can ensure backtrackability if we choose a descent path from the set of feasible ascent paths, such that the robot can always return to its anchor point at the top. If there is no feasible path in this sleeve, we add this sleeve's \mathcal{H} -value to the list of constraints and continue the A* search in regions of different \mathcal{H} -values. This continues until we find a feasible path, or terminate without finding any path.

The ascent path feasibility check remains much the same as that given by Abad-Manterola et al., but is described here as a separate function for clarity. For each anchor point on the SHP, we construct the anchor reachable set based on the terrain and the robot's capabilities. If robot can move so as to remove its tether from contact with the obstacle at that anchor point, that anchor point is considered passable. A series of passable anchor points means that the robot can reach a series of positions that allows it to ascend. If one of the anchor points is not passable, we add this sleeve's \mathcal{H} -value to the list of constraints, and return to the main block of code to continue looking for feasible sleeves.

Algorithm 11 Offline Pre-Planning Algorithm

```

procedure PREPLAN( $v_s, v_g, B$ )
   $\mathcal{H}_c \leftarrow \emptyset$  ▷ Initialize list of  $\mathcal{H}$ -value constraints
   $updates \leftarrow \emptyset$ 
   $feasiblePaths \leftarrow \emptyset$ 
5:  $\pi \leftarrow \emptyset$ 
  while not  $feasiblePaths$  do
    while not  $\pi$  do
       $\pi = A^*(B, \mathcal{H}_c, 0)$  ▷ Run A* with the given constraints, and no
updates
      Find  $S(\pi)$  and its  $\mathcal{H}$ -value
10: Find  $\bar{\pi}$  and identify anchor points  $A$ 
      if CHECKFEASIBILITY3( $\bar{\pi}$ ) then
        Find  $feasiblePaths$  ▷ Pick a pair of ascent/descent paths that pass
through all the anchor reachable set

```

Algorithm 12 Check if SHP is Feasible

```

procedure CHECKFEASIBILITY3( $\bar{\pi}$ )
  Find list of anchor points,  $A$ 
   $continue \leftarrow \text{True}$ 
   $q \leftarrow g$ 
5:  $j \leftarrow \text{length}(A)$ 
   $pathExists \leftarrow \text{False}$ 
  while  $continue$  do
    Construct  $C_j(q)$  ▷ Construct anchor reachable set
     $a_j \leftarrow A[j]$ 
10: if  $a_j$  is passable then
       $j \leftarrow j - 1$ 
      if  $j == 0$  then
         $continue \leftarrow \text{False}$ 
         $pathExists \leftarrow \text{True}$  ▷ Feasible ascent path exists
15: else
       $q \leftarrow \text{SHP edge of } C_j(q)$ 
    else
       $\mathcal{H}_c \leftarrow \{\mathcal{H}_c, \mathcal{H}\}$  ▷ Add  $\mathcal{H}$ -value to list of constraints
       $continue \leftarrow \text{False}$ 
20: return  $pathExists$ 

```

3.5.2 Adjusting the BTM

Assume that the robot has carried out the aforementioned offline planning, selecting a sleeve and its associated SHP, and an ascent/descent path pair. However, as it starts down the prescribed descent path, it receives new information, indicating that the original map is in error. A simple analysis shows that all of the possible errors fall into one or more of the following categories:

1. An entirely new, unforeseen, obstacle appears.
2. A map obstacle is found to be a phantom (i.e., the obstacle “disappears”).
3. An obstacle has shifted location.
4. The boundaries of an obstacle or of the planning region are found to be different from those used to construct the preliminary plan.

Each of these changes will require a modification of the boundary triangulation of the descent slopes. Can this modification be handled in an online fashion? The answer to this question depends upon how changes in obstacle and terrain geometry affect the boundary triangulation. The classes of errors described above can be reduced to two simple classes of modifications: any error can be accommodated by adding and removing obstacles from the map — an existing obstacle or terrain boundary that has changed shape or position can simply be removed and replaced by a new, slightly different obstacle. Therefore, let us assume that we are changing BTMs only to reflect added or removed obstacles.

As proven in Lemma 3.5.1 below, the effects of bounded map errors can be localized to a small region, which we term the *affected region*. This region consists of all the BTM triangles that contain all or part of the added or removed obstacle components. Since the boundary of the affected region consists of edges and vertices, which themselves must be boundary segments due to the properties of the original boundary triangulation, the affected area can be retriangulated as a BTM.

Lemma 3.5.1 *Any boundary triangulated 2-manifold (BTM) can be locally retriangulated in the affected region around a new or removed obstacle to construct a proper boundary triangulation (see Subsection 2.2.2) which seamlessly meshes with the boundary triangulation outside the affected region. The resulting triangulation is a BTM.*

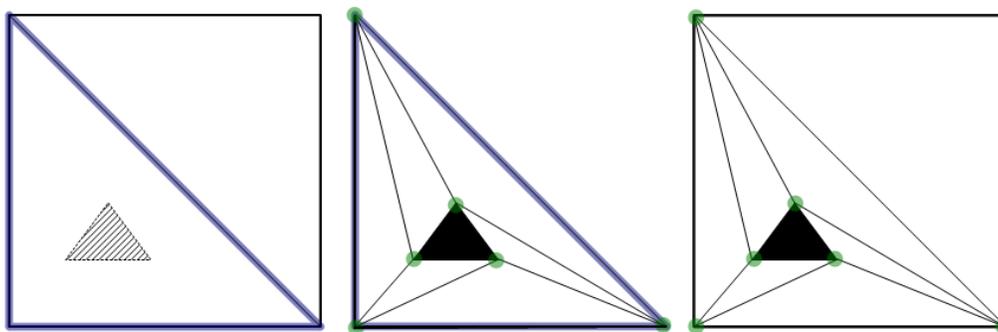


Figure 3.16: Left: New obstacle added to BTM. Center: Triangulate affected area. Right: New BTM.

Proof: We assume that a new obstacle added to the map, or the obstacle to be removed from the map, is a bounded polygon. First consider the case of an added polygonal obstacle. This polygon will intersect a finite number of triangles in the boundary triangulation. The union of these triangles is termed the *affected region*, and is naturally a polygon. By the properties of boundary triangulation (Subsection 2.2.2), the vertices bounding the affected region are boundary vertices. As such, when a polygonal hole representing an obstacle is added to the BTM, a retriangulation of the affected region (incorporating the presence of the new obstacle) using de Berg's algorithm [89] results in a BTM of the affected region. Since the vertices of the affected area are, by definition, boundary vertices of the overall map, the retriangulation within the affected region maintains the BTM requirements for the overall map. Similarly, if an obstacle is removed, let the affected region consist of the union of the triangles whose edges or vertices are incident on the polygonal hole formed by the removed obstacle. Again, the boundary of the affected region satisfies the properties of a boundary triangulation, and thus the affected region can be retriangulated to form a BTM which naturally integrates with the existing BTM. \square

Figure 3.16 illustrates how the process of adding an obstacle affects the triangulation. In the left panel, the new triangular obstacle added to the existing BTM lies wholly within a triangular simplex. That triangle is the affected region, and it is triangulated to form a local BTM (center diagram). Since all of the boundary vertices of this affected region are also boundary vertices of the larger map, the union of the retriangulation and the unaffected region forms a BTM (right panel of Figure 3.16). The case of an obstacle falling into multiple simplices is illustrated in Figure 3.17. The two triangles incident to the obstacle become the affected region, and the entire

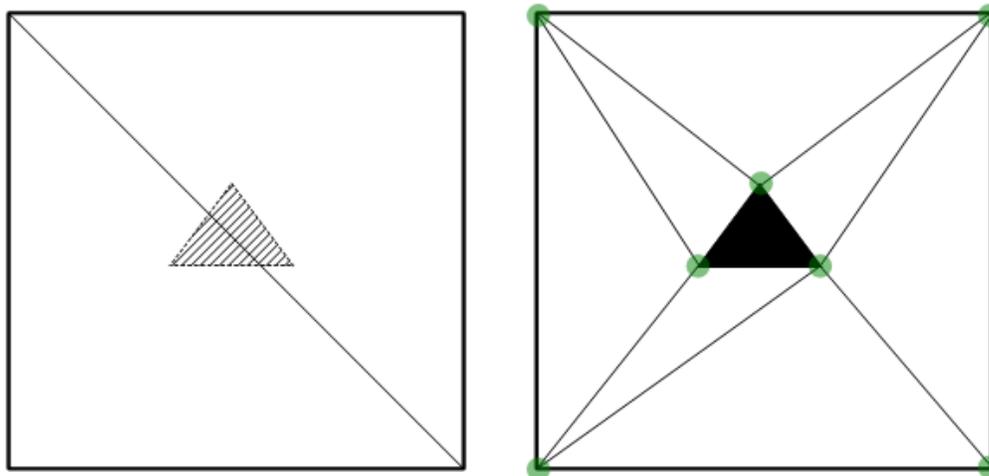


Figure 3.17: Left: New obstacle added to BTM. Right: Retriangulate affected area for new BTM.

square containing both simplices, as well as the newly discovered obstacle, are retriangulated (right-hand diagram of Figure 3.17). As in Figure 3.16, the overall BTM remains a BTM, due to this specific retriangulation of the affected region.

The case for removing an obstacle is much the same, as can be seen by reading Figure 3.16 and Figure 3.17 backwards. The affected region consists of the removed obstacle plus its adjacent simplices. The affected region is retriangulated to form a BTM, which conserves the BTM of the larger map.

3.5.3 Adjusting the Sleeve and Paths

Let the *current sleeve* denote the sleeve in which the robot's evolving descent path lies. When an unexpected obstacle is encountered, it may affect the geometry of the current sleeve, or it may not. In the most drastic effect, the current sleeve can be split into two sleeves, which will have coincident simplices up to the affected region. If there is no effect on the current sleeve, then the original plan can be continued until the next map error is encountered. The retriangulation may be useful later if the robot needs to backtrack.

If the newly observed obstacle intersects the current sleeve, it can have one of several effects:

1. The current sleeve is blocked. The robot must ascend until it finds a junction with an as-yet unexplored sleeve.

2. The sleeve geometry may be altered, but it does not alter the shortest homotopic path. In this case, there are no homotopy effects due to the map errors. The original plan will not be altered unless the obstacle blocks access to terrain whose trafficability is necessary for robot safety and controllability on the ascent or descent path.
3. The sleeve is not split into two pieces, but the shortest homotopic path is altered. The robot must reassess the feasibility of the new SHP. If the path is feasible, the robot can continue. Otherwise, the robot must backtrack to a junction with an as yet unexplored sleeve.
4. The current sleeve is split into two sleeves. The two sleeves will contain the same simplices up until the affected region. In the affected region, the new obstacle will force a *leftward* and *rightward* (with respect to downward movement) sleeve around the obstacle. The robot must then find and evaluate the feasibility of the SHPs for each sleeve.

Similarly, removing an obstacle might free up a new sleeve, but it could also change the anchor points to make a path infeasible. Therefore, we must check for path intersection and feasibility. Algorithm 13 describes when to recompute sleeves and feasibility.

Every map change invokes a local retriangulation — this is a computationally cheap operation. We ignore those changes that happen outside the current sleeve, placing them in a list of updates to be propagated when necessary. For those changes that affect the current sleeve, we update the sleeve to reflect the new triangulation, and recompute the SHP if necessary. If the SHP has changed, then the paths have been materially altered and we must replan from the affected region down to the goal. If there is no feasible path forward, this sleeve is marked as as infeasible by adding its L-value to the list of constraints. Similarly, if the SHP remains the same, but an obstacle blocks a path or comes between the ascent and descent paths, we must select new paths or block A^* in that sleeve by adding that new L-value constraint. Note that L-values are only computed as A^* expands that node.

We then run the A^* search algorithm again, propagating map updates as we go. Having found a promising sleeve, the robot backtracks to the junction of the two sleeves, and starts down the new sleeve. If there is a feasible path, we must again select an ascent and descent path. By selecting a descent path within the set of

Algorithm 13 Online path planning algorithm

```

procedure ONLINE PLANNING
  updates  $\leftarrow \emptyset$ 
  repeat
    if mapChanged then
5:      updateBTM()
      updates  $\leftarrow \{updates, mapChanges\}$ 
      if affected area is in the current sleeve then
         $\bar{\pi}_{old} \leftarrow \bar{\pi}$ 
        update sleeve around  $\bar{\pi}$ 
10:     recompute  $\bar{\pi}$  and anchor points
        if  $\bar{\pi} \neq \bar{\pi}_{old}$  then
          if not CHECKFEASIBILITY3( $\bar{\pi}$ ) then
             $\mathcal{H}_c = \{\mathcal{H}_c, \mathcal{H}\}$   $\triangleright$  Add  $\mathcal{H}$ -value to constraints
             $A^*(\mathcal{H}_c, update)$ 
15:         Backtrack to go down that sleeve
            select ascent and descent paths
          else
            if Obstacle blocks ascent or descent paths or  $\mathcal{H}_{ascent} \neq$ 
 $\mathcal{H}_{descent}$  then
              if CHECKFEASIBILITY3( $\bar{\pi}$ ) then
20:                 select new ascent and descent paths
              else
                 $\mathcal{H}_c = \{\mathcal{H}_c, \mathcal{H}\}$ 
                 $A^*(\mathcal{H}_c, update)$ 
                Backtrack to go down that sleeve
25:     until goal reached
  
```

feasible ascent paths, we ensure that the robot can always safely backtrack to its starting position.

This algorithm isn't optimal; because it doesn't recompute sleeves and paths globally, a removed or moved object may open up a more optimal sleeve, but the robot prefers to stay in the current sleeve when possible. However, it will eventually find a path to the bottom, if there is one.

3.5.4 Multiple Changed Obstacles

Numerous simultaneous updates to the map do not significantly change the online path planning algorithm. The updates can be processed in sequence by simply inserting a *for loop* before line 6 in Algorithm 3, with an end after line 28, running the intervening code for every affected region.

3.5.5 Update for Removed PTFOs

With the understanding of how to update a BTM or sleeve for a removed obstacle, Algorithm 14 becomes obvious. First, find the obstacle that the slipping anchor p is part of. Since the tether will slip over p , leave that vertex off the list of that obstacle's vertices, essentially clipping off a triangle. If the tether has slipped over all the other vertices of that obstacle, that obstacle will disappear entirely. Either way, the BTM must then be retriangulated to account for the difference in or disappearance of that obstacle. The new sleeve S can be found by tracing the old SHP through the new BTM.

Algorithm 14 Update BTM and sleeve for slipping over a PTFO

```

1: procedure UPDATEWITHSLIPPED( $p, A, F, \bar{\pi}$ )
2:   Find  $a_{i-1}$  and  $a_{i+1}$ , anchors around  $p$  on  $\bar{\pi}$ 
3:   Find PTFO  $obst = [v_1, v_2, v_3 \dots]$  for which  $p$  was an anchor point
4:   if  $len(obst) > 1$  then
5:      $idx \leftarrow obst.find(p)$  ▷ Clip off  $p$  from  $obst$ 
6:      $obst \leftarrow obst.remove(idx)$ 
7:   else
8:     Remove  $obst$ 
9:   Retriangulate BTM
10:  Find new sleeve  $S$  through which  $\bar{\pi}$  passes
11:   $\bar{\pi}_{part} \leftarrow FUNNEL(a_{i-1}, a_{i+1}, S)$ 
12:   $\bar{\pi}' \leftarrow [\bar{\pi}[0 : a_{i-1}], \bar{\pi}_{part}, \bar{\pi}[a_{i+1} : end]]$ 
13:  Find new  $A, F$  in  $\bar{\pi}'$ 
14:  return ( $A, F, \bar{\pi}'$ )

```

At that point, one could run the Funnel Algorithm from Chapter 2 on the entire sleeve, from v_s to v_g . That isn't necessary, though. Instead, find the portion of the sleeve between the 2 anchor points neighboring p . Use the Funnel Algorithm to find the shortest path between them, and then concatenate this with the ends of the old SHP. This new SHP is returned, along with the possibly-changed list of anchors and friction anchors, A and F .

3.6 Summary and Discussion

This chapter has developed methodologies to support tethered motion planning with the complication of frictional tether-terrain interaction. If the tether sticks on the terrain, it forms a tether friction obstacle (TFO), and any terrain object where this is a possibility is called a PTFO. In order to identify PTFOs, the terrain was modeled as a locally-smooth surface. Analysis of its principal curvatures at every point determined where that surface was convex.

The tether was modeled as a frictional rope, represented by an arc-length parameterized curve $\mathbf{x}(s)$ along its centerline. Section 3.2 reviews the rope physics literature in which Maddocks and Keller and Konyukhov derived the conditions necessary for the rope to stay in static equilibrium, using a Coulomb friction model. A given tether configuration on a known terrain with known tension can then be evaluated to determine if that tether will stick. Section 3.3 showed how to compute tension for tethers in static equilibrium or how to check for sticking in tethers with known tension, and gave a few examples. Section 3.4 combined these static equilibrium checks with the tethered planning algorithm in Chapter 2, resulting in an algorithm for determining if a taut tether will stick on a PTFO, and for planning or replanning accordingly.

If a tether will slip on a PTFO, it forces the BTM to be retriangulated; to the planner, that obstacle has suddenly disappeared. Section 3.5 describes how to adjust a BTM, sleeve, and path to account for map changes, resulting in an algorithm for online tethered motion planning. These concepts allowed me to present in Subsection 3.5.5 the last piece necessary to complete Algorithm 7.

3.6.1 Observations on Modeling Assumptions

Given both the size of the tribology research community and the amount of research still being done on the mechanics of rope and ribbon, it is perhaps not surprising that frictional tethers are complicated and difficult to model. In order to make the problem tractable, I made a number of assumptions in this chapter. However, each of those assumptions introduces another layer of uncertainty and thus another potential point of failure, each of which would pose a concern for any potential flight mission. While none of these limitations constitute a project-ending hazard, they should be carefully considered and all reasonable steps taken to mitigate the risks.

Risk Mitigation via a Conservative Model

Using a Coulomb friction model and assuming a known friction coefficient were perhaps the most impactful assumptions made in this work, since all the tether-sticking calculations depend on friction. In reality, the friction coefficient for an unknown regolith on a far-away planet can be difficult to estimate. However, Algorithm 7 can easily be altered to account for a range of friction coefficients.

As described in Subsection 3.4.1, PTFOs are classified as TFOs if $T_{max} < T_{min}e^{\int_s \omega ds}$. If $T_d + \epsilon \not\leq T_d e^{\int_s \omega ds}$, they are ignored as terrain which the tether can slide over. If

neither is true, the PTFO's status depends on the tether tension and configuration. These PTFOs are discarded by the algorithm as being too unpredictable.

To ensure that the algorithm works over a range of friction coefficients, simply use the minimum μ in the calculation of ω^2 in the first equation. This is testing that the tether will stick on an obstacle when the tether is at the lowest possible tension on one side, and the highest possible tension on the other, with the lowest possible friction coefficient. Similarly, use the maximum possible μ in calculating the second equation. This ensures that, even in the highest-friction case, the tether will still slide smoothly over non-obstacle terrain. The result of this conservative alteration is to place more PTFOs in the category of unstable obstacles to be avoided. It may result in fewer feasible paths for the robot to travel, but in return, all of those paths will be stable despite any friction-modeling errors.

Operational Risk Mitigation Strategies

Other modeling assumptions bear further investigation in future work. Until that time, risk from the following assumptions can be mitigated in operation and design:

- The contact mechanics between the tether and the terrain are straightforward. I've assumed the terrain acts as a solid, cohesive mass; in reality, a heavy or tensioned tether on loose granulated material might sink into that substrate. The resulting higher friction would prevent that tether from slipping. The model is conservative with regard to contact mechanics because it would never predict a TFO where none exists. However, the tether might stick in unexpected places. In practice, this could be avoided by maintaining higher tether tension.
- The robot has perfect knowledge of the terrain. This includes knowing the size, shape, and elevation of any potential obstacles. In reality, small obstacles or protrusions on a larger obstacle may not show up on a low-resolution DEM taken from orbit.
- The robot has perfect knowledge of its tether's position, and a reasonably accurate estimate of its tension. Sticking is very dependent on friction and tether position, so even a small error in any of these estimations could cause Axel to unexpectedly lose a friction anchor and slip.

- Axel and its terrain are quasi-static. This means that there is no wind or shifting rocks, and any dynamic tether loads or waves across the tether caused by Axel's movements are ignored. In reality, the caster arm and tether's perturbations while driving may cause Axel to lose a friction anchor, especially as Axel draws near that anchor.
- The tether is weightless, perfectly flexible, and inextensible. The weightless assumption is reasonable for short distances but becomes less so for long traverses, where the tether's inertia becomes significant. All ropes in the air form a catenary curve, which must be accounted for when the tether leaves the ground.

While some of these problems may be solved in future work (which I discuss in Chapter 4), there are a number of possible mitigation strategies that can be used in the interim. Every tether strength or weight improvement makes the weightless assumption more accurate. On the design side, the tether management system has already made Axel more robust by acting as a shock absorber for sudden falls. In addition, there are several software and hardware improvements that could give the algorithm or operator a more accurate estimate of the tether tension and position: “smart” tethers that return curvature or tension information, tether-position estimators or anchor-estimators, tension-measuring sensors, or even a tension estimate based on the output of any motors opposing it.

Operationally, the simplest mitigation strategy is to keep a human in the loop, using the algorithms in this chapter as tools to advise, but also applying common sense and intuition. More low-level autonomy might allow the robot to gracefully handle minor errors, occlusions, or inconsistencies in data, or to recover gracefully from an unexpected slip or lost anchor. The team has previously discussed creating a set of motion primitives that could be useful when the robot gets “stuck”: lifting the caster arm to unstick the tether, rolling back and forth to get out of a rut, or pushing down with the caster arm to climb a large rock unassisted.

Alternately, an operator may choose to reduce the risk of a slipped anchor by avoiding PTFOs as much as possible, choosing a path where the SHP relies only on geometric obstacles. This may be the most reliable strategy, but it also loses some of the potential benefits of friction anchors. As was mentioned in Subsection 3.1.1, using a friction anchor on a gentle slope may allow Axel to move farther laterally on a steeper slope, below. Although avoiding PTFOs entirely might reduce risk,

it would come at enormous cost to the mobility and operation flexibility for which Axel was designed.

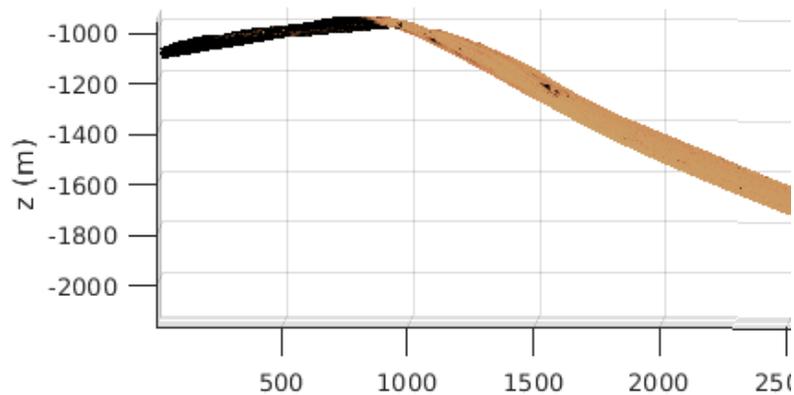


Figure 3.18: The lip of Andapa crater, from its DEM. With a slight upward slant, it's ideal for leveraging the capstan effect to allow a weak anchor holding force.

Tether friction played a huge role in Axel's design and will continue to matter in its operations. Axel's configuration, with the onboard reel paying out tether behind it, was specifically designed to minimize wear and tear on the tether. It also takes advantage of the capstan effect. For example, Axel could set a very weak initial anchor at the top. If it then wrapped the tether around a few more anchors before dropping onto a steeper slope, it can be assured that the initial anchor will hold under even very high Axel-side tension. This can be an effective strategy even without obstacles. As Attaway shows, laying a rope over a cliff edge like the one in Figure 3.18 will magnify the holding force, allowing a weak anchor to stay secure against higher forces below [75]. Whether it hurts or helps, a tethered robot cannot afford to ignore the effects of friction on its tether.

*Chapter 4***CONCLUSION****4.1 Review**

For rappelling rovers like Axel, tether management is a key issue. To prevent its tether from getting tangled or stuck, the outgoing and return paths of any tethered rover must be homotopic. My thesis contributed to our understanding of tether management by showing how shortest-path algorithms from computational geometry can be combined with graph search to quickly and efficiently find the shortest paths in all the homotopy classes of a map. I implemented a program that can read in terrain data, detect tether obstacles, triangulate a map between those obstacles, and search that map for the shortest path or paths between start and goal points. It can then check that class of paths to ensure that Axel will be able to ascend, and generate waypoints for ascent and descent paths. Results from this program are shown in Section 2.5.

While the algorithm and results from Chapter 2 assumed a frictionless tether, Chapter 3 revisited this assumption. I defined tether friction obstacles, which result when a frictional tether sticks to convex terrain, and used existing force equations for static equilibrium in a rope to identify the conditions under which a tether friction obstacle might form. My thesis contributed an algorithm and the tether-tension assumptions necessary to find a path and check if the tether will catch on any tether friction obstacles. This algorithm describes how to alter the path as a result of the dynamic reclassification of PTFOs as TFOs, non-obstacles, or dangerous PTFOs. If it determines that the resulting path is unsafe or infeasible, it replans.

4.2 Future Work**4.2.1 Limitations and Assumptions**

As detailed in Section 3.6, the complexity of the tethered planning problem compelled me to make a number of assumptions, especially where friction was involved. Chapter 2 relies on the assumption that the terrain is known to the appropriate level of detail. The tether is modeled as an idealized weightless inextensible rope, which is a reasonable approximation over short distances.

Revisiting the aforementioned assumptions could improve the algorithm's precision

and robustness. For example, how should incomplete and poor-resolution terrain data be handled? The Garni Crater DEM used in Section 2.5 had a resolution of approximately 1m, which is insufficient for detecting many potentially problematic tether obstacles. However, an on-site autonomous robot could perform online mapping, incorporating new sensor data into that larger satellite map. Even if the terrain data is available at a small enough scale, it is still pixelated. All of the existing tether physics equations are for continuous surfaces.

The terrains in Section 2.5 were treated as flat planes, and obstacles defined by their difference from that plane. Ideally, though, the tether's status could be analyzed against the original surface. This could be accomplished by smoothing out the surface, perhaps using splines. Alternately, it may be possible to translate the existing tether statics equations into their discrete differential geometry counterparts. Doing so would allow the pixelated terrains to be used without modification.

While an idealized rope is a decent approximation of the tether over short distances, over long distances the tether's weight matters. While a weightless rope could easily be pulled forward along flat ground, a heavier rope — or a long section of relatively light tether — could not. Although the current algorithm only deals with a convex series of flat planes, later versions may extend to the full 3 dimensions. In that case, a tether hanging between 2 points forms a catenary curve, another example of how a real-life tether differs from the straight line of the weightless ideal.

Real tethers are not perfectly flexible, either. Axel's tether has a minimum recommended bend radius; if the tether bends too sharply under pressure, it may break. The existing algorithm could be modified to constrain acceptable paths to those that do not exceed the minimum bend radius by disqualifying any paths with too-sharp obstacle geometry near an anchor.

The relative stiffness of Axel's tether means that it does not tend to kink or twist like other cables. Its wire core makes it inextensible. However, more elastic tethers would better protect a climbing robot from the shock of an arrested fall. In this case, the existing rope model could be extended to better model real tethers.

Throughout this work, the robot is approximated as a point mass, and both robot and terrain are assumed to be quasi-static. These approximations are reasonable for a small rover moving slowly around a sparse obstacle field. The tethered planning algorithm produces a homotopy class of paths, providing a general guide for the robot. The robot's exact motions should be chosen in a more precise way, such as

planning in a configuration space that accounts for its dimensions or optimizing its poses on a point-cloud-based map [99]. Similarly, a more sophisticated feasibility check could be implemented using a non-holonomic robot model to account for wheel constraints.

At the moment, geometric tether obstacles are all assumed to be robot obstacles, although in practice there may be obstacles that the robot can climb and the tether cannot. Ultimately, better performance could be obtained if the two sets of obstacles could be detected separately with different models, since the tether and the robot have different limitations. It may also be useful to revisit the assumption that obstacles are polygonal and symmetric, and determine how relaxing these assumptions could change the available solutions.

Chapter 3 adds the assumption that some approximation of the tether tension and configuration are known. The tether-terrain interactions are simplified to a Coulomb friction model, i.e., with a known friction coefficient, on solid ground. The accuracy of the existing algorithm could be improved by adding a more nuanced friction model, and using a terramechanics-based ground contact model.

4.2.2 Other Suggested Improvements

There are a number of potential improvements to the existing software implementation related to usability and design. At present an operator inputs a start point and desired goal. Sometimes, though, a science team is interested in a series of goals. Determining the best start point to reach a goal or goals, and allowing the program to plan paths between multiple goals would make the program easier to operate.

Algorithms 7 and 13 in Chapter 3 handle planning with a frictional tether and online planning, respectively. However, their common basis of addressing possible changes to the obstacle map means they could be combined into an online frictional-tether planning algorithm. Most of the algorithms computed in this thesis will loop forever if a suitable path is not found. This is because there are an infinite number of homotopy classes for any non-simple polygon, with ever-higher numbers of loops. A good next step would be to define some cutoff conditions, such as constraining the algorithm to only return simple homotopy classes.

4.2.3 Fully 3-D Surfaces

Although my work in this thesis has modeled the terrain as a series of planes, it should be possible to use a truly 3-dimensional surface. First, assuming a weightless and

frictionless tether, one could determine where a tether will leave a concave surface, and what path it will take to the next contact point. This would allow tethered motion planning on any series of planes, regardless of their relative angles.

Furthermore, Hershberger and Snoeyink's theory applies to anything that can be represented as a BTM. This should allow the use of 3-D triangle chains which can be isometrically flattened, as in the 3-D geodesic literature [92], [91]. Since SHPs on a convex surface are geodesics, it may be possible to adapt the Funnel Algorithm and SPT Algorithm to truly 3-D surfaces, or to use some of the geodesic-finding algorithms to find SHPs on a 3-D surface.

4.3 Applications

The tethered motion planning algorithm in this thesis can be used to plan Axel's general strategy for descending and ascending a steep slope. A feasible path found by the algorithm could act as a general guide for human teleoperation, to ensure that safe ascent is possible while maintaining the advantages of keeping a human in the loop. If some event causes the robot to halt partway down the slope, the program can offer alternative routes. Alternately, the sleeve or homotopy class for that path can be passed to a more detailed robot motion planner, giving the assurance that the resulting motions will not tangle the tether. In this case, tether and robot motion planners acting together would allow Axel to operate autonomously.

Although this tethered planning algorithm was designed with Axel in mind, it can be applied to many other tethered climbing robots. Dante II, discussed in Section 1.3, could use it to plan a new route into a volcanic caldera without tipping over. A robot descending into a mine or scouting a cave could use its sensors to map terrain and plan a feasible ascent/decent path pair to ensure that it can autonomously find its way out without human intervention. By autonomously charting decent and ascent paths, Forest Service robots could automatically place sensors on mountains to check for potential rock slides or avalanches. Mining robots could probe the stability of the sides of open-pit mines, without risking human lives.

There are many other reasons for robots to use tethers, as discussed in Section 1.4. Any ground-traveling robot with a tether, cable, hose, or other flexible rope-like structure could use the method described in Subsection 2.2.4 for motion planning in a tether-free plane. This includes robots using a cable for power or communication, such as a one entering a radiation-heavy disaster zone. Autonomous tethered planning capability would allow that robot to make faster decisions without human

input, potentially freeing up human operators to help with the disaster response. This algorithm may also prove useful for other robots dealing with ropes, like an ROV traversing the ocean floor to lay fiber optic cable, or a robot using a hose to water a garden. In general, such a robot should constrain its motions to outgoing and incoming path pairs that are homotopic, however that is achieved.

Although the algorithm described in Chapter 2 is most efficient for Axel's use, other motion planning architectures might favor checking 2 generated potential paths for homotopy, or by otherwise applying the algorithms and principles outlined in this thesis. One such application would be the case when 2 robots are connected together as they move — a robot towing a vehicle, or two robots dragging a net between them. In either case, the two elements must take homotopic paths, and stay within the length of the connecting tether. Even JPL's TRESSA could check potential sites to analyze where it can travel: although its 2 tethers form a closed loop rather than a single path, it must still operate within the homotopy class formed by those 2 tethers.

Even more broadly, homotopy-constrained search may be useful for any applications in which it matters not only that a robot reach its goal, but the path it takes to get there. An autonomous car could use online identification of homotopy classes on a freeway, as a way to generate alternate routes in a different homotopy class. Perhaps a traveling salesman could take the longest simple homotopic path, or a vacuuming robot in an open-floorplan house could ensure that it has covered every homotopy class. With further research, we might someday see tethered drones in the skies, tethered robot vacuum cleaners in our houses, and yes, tethered rappelling robots on other planets.

The tractability of tethered path planning has the potential to make tethered robots more common in our daily lives. Using algorithms based on the work in this thesis, tomorrow's robots will have the ability to plan around tethers in order to water a garden with a hose or hook an electric vehicle up to a charging station. Even the fiber arts might benefit from advances in the field. While large commercial knitting machines and looms require no understanding of homotopy, a smaller robot knitting custom, made-to-order pieces could become feasible with proper extensions to this work. I hope the reader, upon setting down this thesis, will begin to observe the ubiquity of ropes, tethers, and cords in our daily lives. As robots likewise become more common, their interactions with these rope-like objects will almost certainly remain a area of important research, even as the casual observer remains unaware of

the important path planning algorithms being seamlessly executed all around them.

BIBLIOGRAPHY

- [1] J. F. Parker and V. R. West, *Bioastronautics Data Book*, 2nd ed. Washington, D.C.: NASA, 1973. [Online]. Available: <http://adsabs.harvard.edu/full/1973NASSP3006.....P> (visited on 05/07/2018).
- [2] N. JPL. (). “Curiosity Rover Mission Overview,” [Online]. Available: <https://mars.nasa.gov/msl/mission/overview/> (visited on 08/01/2018).
- [3] (). “Voyager - Mission Overview,” [Online]. Available: <https://voyager.jpl.nasa.gov/mission/> (visited on 08/01/2018).
- [4] (). “Cassini-Huygens,” [Online]. Available: <http://www.jpl.nasa.gov/missions/cassini-huygens/> (visited on 08/01/2018).
- [5] N. C. Administrator. (Aug. 7, 2017). “Cape St. Vincent,” [Online]. Available: http://www.nasa.gov/multimedia/imagegallery/image_feature_873.html (visited on 07/31/2018).
- [6] S. C. for NASA Technology Roadmaps, *NASA Space Technology Roadmaps and Priorities: Restoring NASA’s Technological Edge and Paving the Way for a New Era in Space*. The National Academies Press, 2012, ISBN: 978-0-309-25362-8.
- [7] “NASA Technology Roadmaps - TA 4: Robotics and Autonomous Systems,” *Robotics and Autonomous Systems*, p. 188, 2015.
- [8] G. Visentin, D. Noelke, M. Delpeche, S. Moreno, R. Bertacin, J. Rodriguez, and D. Jones, *Strategic Research Cluster: Space Robotics Technologies. Guidance Document for Horizon 2020 Work Programmer 2018-2020*.
- [9] (Jun. 12, 2017). “Extreme Planet Takes Its Toll,” [Online]. Available: <https://mars.nasa.gov/mer/spotlight/20070612.html> (visited on 08/01/2018).
- [10] N. Barlow, *Mars: An Introduction to Its Interior, Surface and Atmosphere*. Cambridge: Cambridge University Press, 2008, ISBN: 978-0-511-53606-9. doi: 10.1017/CB09780511536069. [Online]. Available: <http://ebooks.cambridge.org/ref/id/CB09780511536069> (visited on 08/01/2018).
- [11] A. S. McEwen, L. Ojha, C. M. Dundas, S. S. Mattson, S. Byrne, J. J. Wray, S. C. Cull, S. L. Murchie, N. Thomas, and V. C. Gulick, “Seasonal Flows on Warm Martian Slopes,” *Science*, vol. 333, no. 6043, pp. 740–743, Aug. 5, 2011, ISSN: 0036-8075, 1095-9203. doi: 10.1126/science.1204816. pmid: 21817049. [Online]. Available: <http://science.sciencemag.org/content/333/6043/740> (visited on 08/01/2018).
- [12] (). “NASA - Oblique View of Warm Season Flows in Newton Crater,” [Online]. Available: https://www.nasa.gov/mission_pages/MRO/multimedia/pia14479.html (visited on 07/31/2018).

- [13] L. Ojha, M. B. Wilhelm, S. L. Murchie, A. S. McEwen, J. J. Wray, J. Hanley, M. Massé, and M. Chojnacki, “Spectral evidence for hydrated salts in recurring slope lineae on Mars,” *Nature Geoscience*, vol. 8, no. 11, pp. 829–832, Nov. 2015, ISSN: 1752-0908. DOI: 10.1038/ngeo2546. [Online]. Available: <https://www.nature.com/articles/ngeo2546> (visited on 08/01/2018).
- [14] A. McEwen, M. Chojnacki, C. Dundas, L. Ojha, M. Masse, E. Schaefer, and C. Leung, “Recurring Slope Lineae on Mars: Atmospheric Origin?” In *EPSC Abstracts*, vol. 10, 2015.
- [15] V. F. Chevrier and E. G. Rivera-Valentin, “Formation of recurring slope lineae by liquid brines on present-day Mars,” *Geophysical Research Letters*, vol. 39, no. 21, 2012, ISSN: 1944-8007. DOI: 10.1029/2012GL054119. [Online]. Available: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2012GL054119> (visited on 12/17/2018).
- [16] D. E. Stillman, T. I. Michaels, R. E. Grimm, and J. Hanley, “Observations and modeling of northern mid-latitude recurring slope lineae (RSL) suggest recharge by a present-day martian briny aquifer,” *Icarus*, vol. 265, pp. 125–138, Feb. 1, 2016, ISSN: 0019-1035. DOI: 10.1016/j.icarus.2015.10.007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S001910351500473X> (visited on 12/17/2018).
- [17] S. Cull, R. E. Arvidson, R. V. Morris, M. Wolff, M. T. Mellon, and M. T. Lemmon, “Seasonal ice cycle at the Mars Phoenix landing site: 2. Postlanding CRISM and ground observations,” *Journal of Geophysical Research*, vol. 115, May 29, 2010, ISSN: 0148-0227. DOI: 10.1029/2009JE003410. [Online]. Available: <http://doi.wiley.com/10.1029/2009JE003410> (visited on 12/17/2018).
- [18] J. T. Wilson, V. R. Eke, R. J. Massey, R. C. Elphic, W. C. Feldman, S. Maurice, and L. F. A. Teodoro, “Equatorial locations of water on Mars: Improved resolution maps based on Mars Odyssey Neutron Spectrometer data,” Aug. 1, 2017. DOI: 10.1016/j.icarus.2017.07.028. [Online]. Available: <https://arxiv.org/abs/1708.00518> (visited on 12/17/2018).
- [19] J. I. Núñez, O. S. Barnouin, S. L. Murchie, F. P. Seelos, J. A. McGovern, K. D. Seelos, and D. L. Buczkowski, “New insights into gully formation on Mars: Constraints from composition as seen by MRO/CRISM,” *Geophysical Research Letters*, vol. 43, no. 17, pp. 8893–8902, 2016, ISSN: 1944-8007. DOI: 10.1002/2016GL068956. [Online]. Available: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2016GL068956> (visited on 12/18/2018).
- [20] C. M. Dundas, A. S. McEwen, M. Chojnacki, M. P. Milazzo, S. Byrne, J. N. McElwaine, and A. Urso, “Granular flows at recurring slope lineae on Mars indicate a limited role for liquid water,” *Nature Geoscience*, vol. 10, no. 12, p. 903, Dec. 2017, ISSN: 1752-0908. DOI: 10.1038/s41561-017-0012-5.

- [Online]. Available: <https://www.nature.com/articles/s41561-017-0012-5> (visited on 12/17/2018).
- [21] G. Meirion-Griffith, I. Nesnas, L. Kerber, R. Anderson, T. Brown, F. Calef, J. Burdick, and M. Tanner, “Accessing Mars recurring slope lineae: Mobility systems analysis,” in *2018 IEEE Aerospace Conference*, Mar. 2018, pp. 1–13. DOI: 10.1109/AERO.2018.8396658.
- [22] P. Mahaffy and J. Ash-Poole. (). “SAM: Sample Analysis at Mars on the rover ‘Curiosity’,” [Online]. Available: <https://ssed.gsfc.nasa.gov/sam/curiosity.html> (visited on 12/19/2018).
- [23] H. Hiesinger, J. W. Head, U. Wolf, R. Jaumann, and G. Neukum, “Ages and stratigraphy of mare basalts in Oceanus Procellarum, Mare Nubium, Mare Cognitum, and Mare Insularum,” *Journal of Geophysical Research: Planets*, vol. 108, no. E7, 2003, ISSN: 2156-2202. DOI: 10.1029/2002JE001985. [Online]. Available: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2002JE001985> (visited on 12/20/2018).
- [24] J. Haruyama, K. Hioki, M. Shirao, T. Morota, H. Hiesinger, C. H. van der Bogert, H. Miyamoto, A. Iwasaki, Y. Yokota, M. Ohtake, T. Matsunaga, S. Hara, S. Nakanotani, and C. M. Pieters, “Possible lunar lava tube skylight observed by SELENE cameras,” *Geophysical Research Letters*, vol. 36, no. 21, Nov. 12, 2009, ISSN: 0094-8276. DOI: 10.1029/2009GL040635. [Online]. Available: <http://doi.wiley.com/10.1029/2009GL040635> (visited on 06/04/2018).
- [25] S. Matthewson. (Jan. 18, 2018). “Lunar Fountain? Accessible Ice Could Lurk in Moon’s Lava Tubes,” [Online]. Available: <https://www.space.com/39404-lava-tube-skylights-discovered-moon-images.html> (visited on 12/20/2018).
- [26] L. Kerber, I. Nesnas, J. W. Ashley, M. J. Malaska, C. Parcheta, K. L. Mitchell, and R. C. Anderson, “Moon Diver: A Mission Concept for Exploring the History of Lunar Mare Deposits with the Axel Extreme Terrain Rover,” in *Annual Meeting of the Lunar Exploration Analysis Group*, ser. LPI Contributions, vol. 1960, Nov. 2016, p. 5068.
- [27] J. L. Margot, D. B. Campbell, R. F. Jurgens, and M. A. Slade, “Topography of the Lunar Poles from Radar Interferometry: A Survey of Cold Trap Locations,” *Science*, vol. 284, no. 5420, pp. 1658–1660, Jun. 4, 1999, ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.284.5420.1658. pmid: 10356393. [Online]. Available: <http://science.sciencemag.org/content/284/5420/1658> (visited on 12/20/2018).
- [28] D. J. Lawrence, W. C. Feldman, R. C. Elphic, J. J. Hagerty, S. Maurice, G. W. McKinney, and T. H. Prettyman, “Improved modeling of Lunar Prospector neutron spectrometer data: Implications for hydrogen deposits at the lunar poles,” *Journal of Geophysical Research: Planets*, vol. 111, no. E8,

- 2006, ISSN: 2156-2202. DOI: 10.1029/2005JE002637. [Online]. Available: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2005JE002637> (visited on 12/20/2018).
- [29] M. T. Zuber, J. W. Head, D. E. Smith, G. A. Neumann, E. Mazarico, M. H. Torrence, O. Aharonson, A. R. Tye, C. I. Fassett, M. A. Rosenburg, and H. J. Melosh, “Constraints on the volatile distribution within Shackleton crater at the lunar south pole,” *Nature*, vol. 486, no. 7403, pp. 378–381, Jun. 2012, ISSN: 1476-4687. DOI: 10.1038/nature11216. [Online]. Available: <https://www.nature.com/articles/nature11216> (visited on 12/20/2018).
- [30] E. Guizzo. (Aug. 23, 2011). “Fukushima Robot Operator Writes Tell-All Blog,” [Online]. Available: <https://spectrum.ieee.org/automaton/robotics/industrial-robots/fukushima-robot-operator-diaries> (visited on 05/12/2019).
- [31] (). “Overview | DRC Finals,” [Online]. Available: <https://archive.darpa.mil/roboticschallenge/overview.html> (visited on 12/20/2018).
- [32] T. Greicius. (). “Rover Takes on Steepest Slope Ever Tried on Mars,” [Online]. Available: <http://www.nasa.gov/feature/jpl/rover-takes-on-steepest-slope-ever-tried-on-mars> (visited on 05/12/2019).
- [33] J. Sessions, B. Leshchinsky, W. Chung, K. Boston, and J. Wimer, “Theoretical Stability and Traction of Steep Slope Tethered Feller-Bunchers,” *Forest Science*, vol. 63, no. 2, pp. 192–200, Apr. 19, 2017, ISSN: 0015-749X. DOI: 10.5849/forsci.16-069. [Online]. Available: <https://academic.oup.com/forestscience/article/63/2/192-200/4583967> (visited on 05/12/2019).
- [34] (). “DARPA Robotics Challenge: Amazing Moments, Lessons Learned, and What’s Next - IEEE Spectrum,” [Online]. Available: <https://spectrum.ieee.org/automaton/robotics/humanoids/darpa-robotics-challenge-amazing-moments-lessons-learned-whats-next> (visited on 05/31/2020).
- [35] J. E. Bares and D. S. Wettergreen, “Dante II: Technical Description, Results, and Lessons Learned,” *The International Journal of Robotics Research*, vol. 18, no. 7, pp. 621–649, Jul. 1, 1999, ISSN: 0278-3649. DOI: 10.1177/02783649922066475. [Online]. Available: <https://doi.org/10.1177/02783649922066475> (visited on 05/21/2018).
- [36] M. Krishna, J. Bares, and E. Mutschler, “Tethering system design for Dante II,” in *Proceedings of International Conference on Robotics and Automation*, vol. 2, Apr. 1997, 1100–1105 vol.2. DOI: 10.1109/ROBOT.1997.614283.
- [37] G. L. Paulsen, S. Farritor, T. L. Huntsberger, and H. Aghazarian, “All Terrain Exploration with the Cliff-bot System,” in *Proceedings of the 2005 IEEE*

- International Conference on Robotics and Automation*, Apr. 2005, pp. 721–726. DOI: 10.1109/ROBOT.2005.1570203.
- [38] P. Pirjanian, C. Leger, E. Mumm, B. Kennedy, M. Garrett, H. Aghazarian, S. Farritor, and P. Schenker, “Distributed control for a modular, reconfigurable cliff robot,” in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 4, 2002, 4083–4088 vol.4. DOI: 10.1109/ROBOT.2002.1014381.
- [39] *Field and Service Robotics: Results of the 9th International Conference*, 1st edition, ser. Springer Tracts in Advanced Robotics 105. New York: Springer, 2014, ISBN: 978-3-319-07487-0.
- [40] P. McGarey, F. Pomerleau, and T. D. Barfoot, “System Design of a Tethered Robotic Explorer (TRex) for 3D Mapping of Steep Terrain and Harsh Environments,” in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, Springer, Cham, 2016, pp. 267–281, ISBN: 978-3-319-27700-4 978-3-319-27702-8. DOI: 10.1007/978-3-319-27702-8_18. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-27702-8_18 (visited on 06/04/2018).
- [41] P. McGarey, D. Yoon, T. Tang, F. Pomerleau, and T. D. Barfoot, “Field Deployment of the Tethered Robotic eXplorer to Map Extremely Steep Terrain,” in *Field and Service Robotics*, ser. Springer Proceedings in Advanced Robotics, Springer, Cham, 2018, pp. 303–317, ISBN: 978-3-319-67360-8 978-3-319-67361-5. DOI: 10.1007/978-3-319-67361-5_20. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-67361-5_20 (visited on 06/04/2018).
- [42] K. Yoshida, N. Britton, and J. Walker, “Moonraker and Tetris: Japanese Microrovers for Lunar Cave Exploration,” *2nd International Planetary Caves Conference*, 2015.
- [43] C. E. Parcheta, C. A. Pavlov, N. Wiltsie, K. C. Carpenter, J. Nash, A. Parness, and K. L. Mitchell, “A robotic approach to mapping post-eruptive volcanic fissure conduits,” *Journal of Volcanology and Geothermal Research*, vol. 320, pp. 19–28, Jun. 15, 2016, ISSN: 0377-0273. DOI: 10.1016/j.jvolgeores.2016.03.006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S037702731630021X> (visited on 05/23/2019).
- [44] T. Greicius. (). “NASA Robot Plunges Into Volcano to Explore Fissure,” [Online]. Available: <http://www.nasa.gov/jpl/nasa-robot-plunges-into-volcano-to-explore-fissure> (visited on 05/23/2019).
- [45] T. Naillon and C. Rappin, *Best Management and Operating Practices for Steep Slope Machine Logging*, Jun. 2019.
- [46] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, “BigDog, the Rough-Terrain Quadruped Robot,” *IFAC Proceedings Volumes*, 17th IFAC World Congress, vol. 41, no. 2, pp. 10 822–10 825, Jan. 1, 2008, ISSN:

- 1474-6670. DOI: 10.3182/20080706-5-KR-1001.01833. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1474667016407020> (visited on 05/31/2019).
- [47] K. Y. Ma, P. Chirarattananon, S. B. Fuller, and R. J. Wood, "Controlled Flight of a Biologically Inspired, Insect-Scale Robot," *Science*, vol. 340, no. 6132, pp. 603–607, May 3, 2013, ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1231806. PMID: 23641114. [Online]. Available: <https://science.sciencemag.org/content/340/6132/603> (visited on 05/31/2019).
- [48] (). "Safe-T | Smart Tethered Drone Station," [Online]. Available: <https://elistair.com/safe-t-tethered-drone-station> (visited on 05/23/2019).
- [49] (). "Powerline Tethered Drone Systems," [Online]. Available: <https://www.ntpdrones.com/> (visited on 05/23/2019).
- [50] (). "Aeromana - Tethered Drone Solutions - Ideal for Surveillance, Live Stream," [Online]. Available: <https://www.aeromana.com/> (visited on 05/23/2019).
- [51] D. Ratner and P. McKerrow, "Aerial tethered robotics system with hovering-hopping agents for security and rescue operations," in *Proceedings of AUVSI International Conference on Unmanned Systems*, North America, 2006.
- [52] P. J. McKerrow and D. Ratner, "The design of a tethered aerial robot," presented at the IEEE International Conference on Robotics and Automation, 2007.
- [53] M. M. Tanner, J. W. Burdick, and I. A. Nenas, "Online motion planning for tethered robots in extreme terrain," in *Robotics and Automation (ICRA), 2013 IEEE International Conference On*, 2013, pp. 5557–5564. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6631375 (visited on 01/01/2014),
- [54] I. A. D. Nenas, J. B. Matthews, P. Abad-Manterola, J. W. Burdick, J. A. Edlund, J. C. Morrison, R. D. Peters, M. M. Tanner, R. N. Miyake, B. S. Solish, and R. C. Anderson, "Axel and DuAxel Rovers for the Sustainable Exploration of Extreme Terrains," *Journal of Field Robotics*, vol. 29, no. 4, pp. 663–685, Jul. 2012–8. DOI: 10.1002/rob.21407. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1002/rob.21407/abstract>.
- [55] J. B. Matthews and I. A. Nenas, "On the Design of the Axel and DuAxel Rovers for Extreme Terrain Exploration," in *Proceedings of the 2012 IEEE Aerospace Conference*, 2012. DOI: 10.1109/AERO.2012.6187039.

- [56] P. Abad-Manterola, J. Burdick, I. A. D. Nesnas, and J. Cecava, “Wheel Design and Tension Analysis for the Tethered Axel Rover on Extreme Terrain,” in *Proceedings of the 2009 IEEE Aerospace Conference*, Big Sky, Montana, Mar. 2009. DOI: 10.1109/AERO.2009.4839308.
- [57] P. Abad-Manterola, J. W. Burdick, I. A. D. Nesnas, S. Chinchali, C. Fuller, and X. Zhou, “Axel Rover Paddle Wheel Design, Efficiency, and Sinkage on Deformable Terrain,” in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, May 2010, pp. 2821–2827. DOI: 10.1109/ROBOT.2010.5509391.
- [58] P. Abad-Manterola, “Axel Rover Tethered Dynamics and Motion Planning on Extreme Planetary Terrain,” California Institute of Technology, 2012.
- [59] P. Abad-Manterola, I. A. D. Nesnas, and J. W. Burdick, “Motion Planning on Steep Terrain for the Tethered Axel Rover,” in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011, pp. 4188–4195.
- [60] D. Tsai, I. A. D. Nesnas, and D. Zarzhitsky, “Autonomous vision-based tethered-assisted rover docking,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 2834–2841. DOI: 10.1109/IROS.2013.6696757.
- [61] K. Shankar and J. W. Burdick, “Motion planning and control for a tethered, rimless wheel differential drive vehicle,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 4829–4836. DOI: 10.1109/IROS.2013.6697053.
- [62] B. Chazelle, “A Theorem on Polygon Cutting with Applications,” in *23rd Annual Symposium on Foundations of Computer Science*, 1982, pp. 339–349. DOI: 10.1109/SFCS.1982.58.
- [63] D. T. Lee and F. P. Preparata, “Euclidean shortest paths in the presence of rectilinear barriers,” *Networks*, vol. 14, no. 3, pp. 393–410, 1984. DOI: 10.1002/net.3230140304.
- [64] L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. E. Tarjan, “Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons,” *Algorithmica*, vol. 2, no. 1-4, pp. 209–233, 1987. [Online]. Available: <http://link.springer.com/article/10.1007/BF01840360> (visited on 02/26/2014).
- [65] J. Hershberger and J. Snoeyink, “Computing minimum length paths of a given homotopy class,” *Computational Geometry*, vol. 4, no. 2, pp. 63–97, Jun. 1994.
- [66] J. S. Mitchell, “Shortest Paths and Networks,” in *Handbook of Discrete and Computational Geometry, Third Edition*, J. Goodman, J. O’Rourke, and C. D. Tóth, Eds., Third, Boca Raton, FL: CRC Press LLC, 2017, ISBN: 978-1-4987-

- 1139-5. [Online]. Available: <http://www.crcnetbase.com/doi/abs/10.1201/9781420035315.ch27> (visited on 09/12/2018).
- [67] P. G. Xavier, “Shortest Path Planning for a Tethered Robot or an Anchored Cable,” ser. Proceedings of the 1999 IEEE International Conference on Robotics and Automation, vol. 2, Detroit, Michigan, May 1999, pp. 1011–1017.
- [68] F. W. Sinden, “The Tethered Robot Problem,” *The International Journal of Robotics Research*, vol. 9, no. 1, pp. 122–133, Feb. 1990.
- [69] S. Hert and V. Lumelsky, “The Ties that Bind: Motion Planning for Multiple Tethered Robots,” *Robotics and Autonomous Systems*, vol. 17, no. 3, pp. 187–215, May 1996.
- [70] ———, “Moving Multiple Tethered Robots between Arbitrary Configurations,” in *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, Aug. 1995, pp. 280–285. DOI: 10.1109/IROS.1995.526173.
- [71] ———, “Planar Curve Routing for Tethered-Robot Motion Planning,” *International Journal of Computational Geometry and Applications*, vol. 7, no. 3, pp. 225–252, 1997. DOI: 10.1142/S0218195997000156.
- [72] ———, “Motion planning in \mathbb{R}^3 for multiple tethered robots,” *IEEE Transactions on Robotics and Automation*, vol. 15, no. 4, pp. 623–639, Aug. 1999, ISSN: 1042-296X. DOI: 10.1109/70.781966.
- [73] J. H. Maddocks and J. B. Keller, “Ropes in equilibrium,” *SIAM Journal on Applied Mathematics*, vol. 47, no. 6, pp. 1185–1200, 1987. [Online]. Available: <http://epubs.siam.org/doi/pdf/10.1137/0147080> (visited on 08/14/2017).
- [74] A. Konyukhov, “Contact of ropes and orthotropic rough surfaces,” *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 95, no. 4, pp. 406–423, Apr. 2015, ISSN: 00442267. [Online]. Available: <http://doi.wiley.com/10.1002/zamm.201300129> (visited on 08/14/2017).
- [75] S. Attaway, “The Mechanics of Friction in Rope Rescue,” presented at the International Technical Rescue Symposium, 1999. [Online]. Available: http://www.jrre.org/att_frict.pdf (visited on 09/13/2013).
- [76] (). “Entry, Descent, and Landing Technologies,” [Online]. Available: <https://mars.nasa.gov/mars2020/mission/technology/entry-descent-landing/> (visited on 03/04/2019).
- [77] J. Erickson, “Shortest Homotopic Paths,” Online Lecture Notes from University of Illinois’ CS 598: Computational Topology (Fall 2009), Online Lecture Notes from University of Illinois’ CS 598: Computational Topology (Fall 2009), [Online]. Available: <http://jeffe.cs.illinois.edu/teaching/comptop/2009/notes/shortest-homotopic-paths.pdf>.

- [78] ———, “Homotopy of Curves on Surfaces,” Online Lecture Notes from University of Illinois’ CS 598: Computational Topology (Fall 2009), Online Lecture Notes from University of Illinois’ CS 598: Computational Topology (Fall 2009), [Online]. Available: <http://jeffe.cs.illinois.edu/teaching/comptop/2009/notes/surface-homotopy.pdf> (visited on 07/22/2015).
- [79] ———, “Homotopy,” Online Lecture Notes from University of Illinois’ CS 598: Computational Topology (Spring 2013), Online Lecture Notes from University of Illinois’ CS 598: Computational Topology (Spring 2013), [Online]. Available: <http://jeffe.cs.illinois.edu/teaching/comptop/chapters/03-plane-homotopy.pdf> (visited on 07/25/2018).
- [80] S. Cabello, Y. Lin, A. Mantler, and J. Snoeyink, “Testing Homotopy for Paths in the Plane,” *Discrete Computational Geometry*, vol. 31, no. 1, pp. 61–81, 2004. [Online]. Available: <http://graphics.stanford.edu/courses/cs468-02-fall/readings/homotopy.pdf> (visited on 01/10/2013).
- [81] A. Efrat, S. G. Kobourov, and A. Lubiw, “Computing homotopic shortest paths efficiently,” *Computational Geometry*, vol. 35, no. 3, pp. 162–172, Oct. 1, 2006, ISSN: 0925-7721. DOI: 10.1016/j.comgeo.2006.03.003. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0925772106000368> (visited on 04/04/2019).
- [82] S. Bespamyatnikh, “Computing homotopic shortest paths in the plane,” *Journal of Algorithms*, vol. 49, no. 2, pp. 284–303, 2003, ISSN: 0196-6774. DOI: 10.1016/S0196-6774(03)00090-7. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0196677403000907>.
- [83] S. Kim, K. Sreenath, S. Bhattacharya, and V. Kumar, “Trajectory Planning for Systems with Homotopy Class Constraints,” in *Latest Advances in Robot Kinematics*, J. Lenarcic and M. Husty, Eds., Springer Netherlands, 2012, pp. 83–90, ISBN: 978-94-007-4620-6.
- [84] S. Bhattacharya, M. Likhachev, and V. Kumar, “Topological constraints in search-based robot path planning,” *Autonomous Robots*, vol. 33, no. 3, pp. 273–290, Oct. 2012, ISSN: 0929-5593, 1573-7527. DOI: 10.1007/s10514-012-9304-1. [Online]. Available: <http://link.springer.com/10.1007/s10514-012-9304-1> (visited on 06/19/2014).
- [85] S. Bhattacharya, M. Likhachev, and V. Kumar, “Search-Based Path Planning with Homotopy Class Constraints,” in *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, Atlanta, Georgia, Jul. 2010.
- [86] S. Kim, S. Bhattacharya, H. K. Heidarsson, G. Sukhatme, and V. Kumar, “A Topological Approach to Using Cables to Separate and Manipulate Sets of Objects,” in *Robotics: Science and Systems*, 2013. [Online]. Available: http://hans.math.upenn.edu/~subhrabh/file_cache/homotopy_caging_RSS13_0dc2a.pdf (visited on 06/19/2014).

- [87] S. Kim, S. Bhattacharya, and V. Kumar, “Path planning for a tethered mobile robot,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference On*, IEEE, 2014, pp. 1132–1139. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6906996 (visited on 07/21/2015).
- [88] S. Kim and M. Likhachev, “Path planning for a tethered robot using Multi-Heuristic A* with topology-based heuristics,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference On*, IEEE, 2015, pp. 4656–4663.
- [89] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd. New York: Springer-Verlag, 2008.
- [90] A. Hatcher, *Algebraic Topology*. Cambridge University Press, 2002, ISBN: 0-521-79540-0. [Online]. Available: <http://books.google.com/books?hl=en&lr=&id=xsIiEhRfwuIC&oi=fnd&pg=PR9&dq=%22Towers+410.+Obstruction+Theory%22+%22Minimal+Cell+Structures%22+%22The+Dold-Thom+Theorem%22+%22The+LoopSpace+of+a+Suspension%22+%22Basepoints+and+Homotopy%22+%22of+Cell+Complexes+519.+The+Compact-Open+Topology%22&ots=ZnQKjk6gx0&sig=vNRDpST5HsH9-gu7VH4D4Ri11vU> (visited on 06/23/2014).
- [91] M. Balasubramanian, J. Polimeni, and E. Schwartz, “Exact Geodesics and Shortest Paths on Polyhedral Surfaces,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 1006–1016, Jun. 2009, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2008.213. [Online]. Available: <http://ieeexplore.ieee.org/document/4604669/> (visited on 03/29/2019).
- [92] J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou, “The Discrete Geodesic Problem,” *SIAM Journal on Computing*, vol. 16, no. 4, pp. 647–668, Aug. 1987, ISSN: 0097-5397, 1095-7111. DOI: 10.1137/0216045. [Online]. Available: <http://epubs.siam.org/doi/10.1137/0216045> (visited on 03/29/2019).
- [93] S. Ghosh, *Visibility Algorithms in the Plane*. New York, NY, USA: Cambridge University Press, 2007, ISBN: 0-521-87574-9.
- [94] L. Guibas, “CS268 Handout #7,” Class Lecture Notes, Class Lecture Notes, [Online]. Available: <https://graphics.stanford.edu/courses/cs268-09-winter/notes/handout7.pdf> (visited on 03/21/2014).
- [95] K. Polthier and M. Schmies, “Straightest Geodesics on Polyhedral Surfaces,” *Discrete Differential Geometry*, 2006.
- [96] K. R. Varadarajan and P. K. Agarwal, “Approximating Shortest Paths on a Nonconvex Polyhedron,”

- [97] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, “Obstacle detection and terrain classification for autonomous off-road navigation,” *Autonomous robots*, vol. 18, no. 1, pp. 81–102, 2005.
- [98] M. P. do Carmo, *Differential Geometry of Curves and Surfaces*, 2nd ed. Prentice-Hall, 2016, ISBN: 978-0-486-80699-0.
- [99] P. Krüsi, P. Furgale, M. Bosse, and R. Siegwart, “Driving on point clouds: Motion planning, trajectory optimization, and terrain assessment in generic nonplanar environments,” *Journal of Field Robotics*, vol. 34, no. 5, pp. 940–984, 2017.

INDEX

affected region, 111, 112
algorithms, 39, 41, 44, 48, 51, 106–108, 110, 115, 116
anchor reachable set, 45
ARS edges., 46
Axel, 12

boundary triangulated 2-manifold, 23

central module, 13
current sleeve, 113

DuAxel, 12

feasible, 47
figures, 2, 4, 6, 9, 10, 12, 19, 21, 28–31, 33, 35, 37, 38, 40, 43, 46, 47, 49, 50,
53–69, 71–77, 84, 85, 89, 95–104, 120
funnel, 35

homologous, 26
homotopic, 19

initial anchor point, 45

Meusnier’s Theorem, 94

osculating plane, 83

passable, 47
passable intermediate anchor point., 47
potential tether friction obstacles, 80
predecessor, 36

semi-planar robot, 14
shortest homotopic path, 20
shortest path tree, 39
sleeve, 31

tether friction obstacle, 81
tether friction obstacles, 79
tether-free, 20